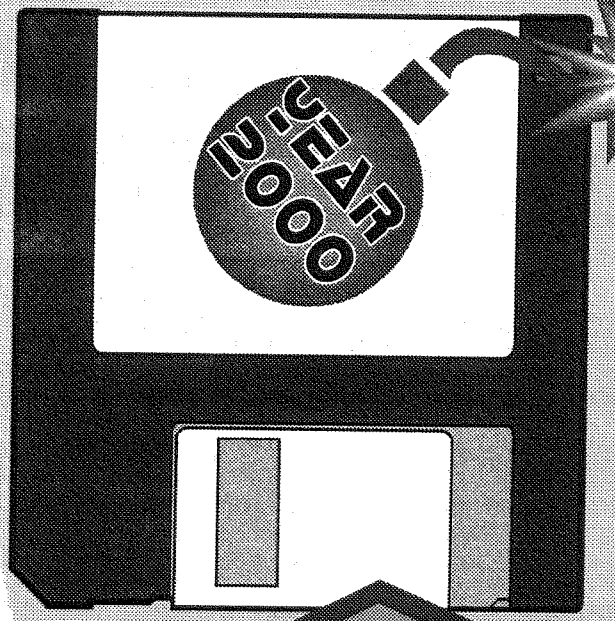
QL Today

Volume 3
Issue 3
Sept./October
1998

ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...

FIND OUT MORE ABOUT THE YEAR 2000 IN THIS ISSUE!



PROBLEM FOR QDOS?

Brandnew: The IBOX!

The new QDOS-based hardware-platform!

Details inside...

Contents

Advertisers

in alphabetical order

Small Ads

Where are your small ads? Searching for something? Want to sell something? 50 words cost only DM 5,- ... 100 words only DM 10,-!

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
email: JMerz@j-m-s.com

English office:

Q Branch
P.O. Box 7
East Sussex
United Kingdom BN41 2ND

Tel. +44 1273 386030
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.
Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque. Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

We welcome your comments, suggstions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 15 April
Issue 2: 15 June
Issue 3: 15 August
Issue 4: 15 October
Issue 5: 15 December
Issue 6: 15 February

Editorial

Dilwyn Jones

It's high summer again - hope everyone who went on holiday had a happy time, and that you're now back happily QLing away! More developments on the operating system front this month. QPC looks set to finally become a native Windows 95 application so that you can switch between tasks on the PC. It's always been the one sore point about QPC for me that you have to spend so long starting it up and shutting it down to change applications. The Q40 is about to gain its first operating system in the form of QDOS-Classic- (formerly Amiga QDOS) by Mark Swift, see his article in this issue. In theory at least, QDOS-Classic- is freely portable to other platforms, as Mark generously makes the sources available. The other emulators are also progressing, and largely thanks to the efforts of their authors in making information about them available via the World Wide Web, we hear of some ex-QL-users returning to the fold as they realise they can use QDOS/SMSQ applications on their favoured hardware. Another interesting development which could lead to QDOS finding application among users of other computers is the proposed new IBOX system from TF Services. Details in this issue, and further details available from Tony Firshman's Web site.

Sadly still no firm news on the 'colour drivers' front, although we are assured that work is proceeding. Major software work always takes a long time, I know. It's a matter of 'when' they'll appear, not 'if' I'm certain!

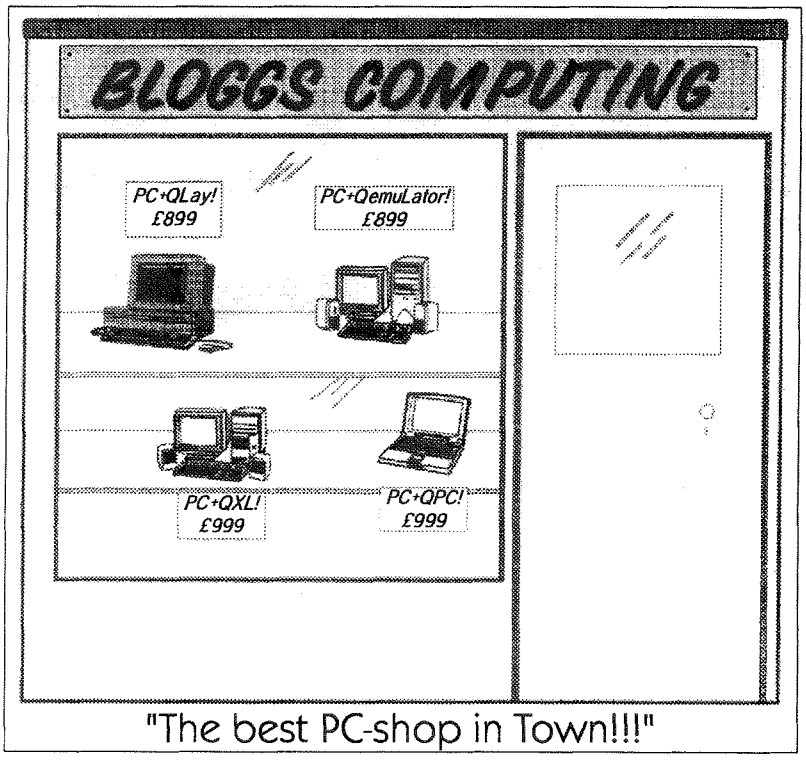
Within this issue we award the first prize for the useful hints competition. Congratulations to Kit Lester - anyone else with any more useful little tips for us to pass on?

I'm having a little get-together of QL friends at my place in the first week of October, just before the Byfleet Quanta workshop. I live near the lovely Snowdonia area of North Wales, so anyone happening to be in the area, or who might like to combine a holiday or a day out with some QLing with Jochen, Marcel Kilgus, myself and a few others would be welcome to join us, get in touch with me first to make arrangements. This is not a formal QL show or workshop, just a few QL users getting together informally over a couple of days.

Norman Dunbar's machine code series moves on to its second instalment in this issue. This major work has been quite well received. I would like to express my gratitude to Norman for agreeing to undertake this major task. We hope you find it useful and worthwhile.

Finally, a little appeal. Our regular contributors do us proud with their articles as ever. How about some contributions from new authors? It doesn't have to be anything particularly advanced, indeed many of our less experienced users enjoy reading beginners' articles. Write and let us know about your QL experiences, or if you have just mastered a particular aspect of your QL, put your experience on paper for other readers to enjoy. Just send the article to us on floppy disk or by email and see your work in print! We'd also like to hear from you if you have ideas on how the QL can be promoted both to non-QL users and to ex-QL users who may wish to start using QDOS or SMSQ again if we can reach them to let them know that it still exists, that there are still plenty of users and traders, that there is still a QL magazine, and that there are now a wide range of QL emulators for other computers.

Cartoon



"The best PC-shop in Town!!!"

NEWS

Darren Branagh

Darren Branagh emailed us to say that the new magazine Z88 USER now has an Email address:
z88user@hotmail.com
Darren himself is also on email now:
darrenbranagh@hotmail.com
He says he'd appreciate emails from British and Irish QL and Z88 users - it gets very lonely where he lives because of the low numbers of QL users in the area!

QL Hackers Journal

Here is the new address for the QL Hacker's Journal:
Timothy Swenson
2455 Medallion Dr.
Union City, CA 94587
Tel. (510) 489-8944
QHJ issue #28 is now out and available on Tim's website **www.geocities.com/Silicon Valley/Pines/5865/**

Steve Johnson Library Closure

Steve Johnson has announced that as of 1st of August 1998, his PD library service (formerly known as SJPD) will finally cease. Steve's library has been a long running source of QL PD, Shareware, Freeware and some Charityware software, and will be sorely missed.
Steve would like to extend his thanks to all his past customers for their support over the years. Likewise, QL Today would like to thank Steve for his hard work and devotion to the QL scene over the years. His service will be greatly missed.
The good news is that Steve is open to offers for the library disks themselves from anyone who would like to buy the entire collection of about 600 disks packed into 2 metal flight cases. This is a wonderful opportunity for someone to acquire an extensive, well organised library of QL software, to set up in business as a source of PD etc software for all QL users.
Offers should be addressed to
Steve Johnson at
36 Eldwick Street,
Burnley,
Lancashire,
England, BB10 3DZ.
Alternatively, Steve can be emailed at
qlpd@johnson.softnet.co.uk

OOPS...

On page 22, in column 2, we inadvertently left a line of German text in the list of printer codes headed "Portrait or landscape". For non-German speaking readers, the translation is: "Please note: Zero and letter O, not double zero"

New QL Web Site

Pedro Reina writes from Spain:
"I have set up a web site, mainly devoted to the QL, so it may be of interest to the QL Today readers. On the web site you can find free software and most of it is QL-related. Of course I would appreciate a visit, and (if possible) a few lines in the News section of QL Today. The URL is:
http://www.anit.es/pedro

RWAP QL Software

I have now updated quite a lot of my software to take account of a problem in the code I use to allow them to load easily from any device - this caused trouble on JM and JS ROMs only.
In conjunction with QBranch, the SBASIC/SuperBASIC Reference Manual has now been released. This is over 1000 pages full of essential information about programming the QL and compatibles.
Work on Q-Route has been proceeding apace and v1.07 is now available, which includes a minor bug fix on the Possible Places menu, as well as an improved display. You can also work in either OS Co-ordinates or the original Q-Route co-ordinates and when you enter the name of a place which needs a county name to distinguish it from other similar places, a separate menu is generated containing only those places with the same name, to choose from.
BRITAIN.MAP was last updated on 6/7/98 - this is available from me or comes supplied with the latest version of Q-Route. Work is now proceeding on a map of Ireland which should hopefully be available from December from me.
I am also making a special offer for all software ordered before 31-10-98 of a 20% discount on all orders over £10 (this excludes the SBASIC/SuperBASIC Reference Manual).
I have now also released some software previously sold by Talent and Microdeal (having updated the programs to work on modern systems and removing copy protection, as well as correcting some bugs):
West and The Lost Kingdom of Zkul (two real-time text adventures) Stone Raider II (an entertaining boulderdash clone) Nemesis MKII (a good text adventure - previously only available as an upgrade from me) Horrorday and the Prawn (two very funny spoof adventures - text only)
3D Terrain has been updated to allow it to work on any screen resolution and also accept export file names in upper case.
Return to Eden has now been upgraded to ensure that it works correctly on all QL ROM versions. It is also easier to EXAMINE certain objects, and there have also been some minor improvements to the interpreter.
FlashBack SE v2.02 is available

from me as an upgrade to the original. This takes into account all of the last final tweaks made to the original program and also ensures that it works on Minerva, SMSQ/E and under the Pointer Environment. The manual has been updated. Finally, I shall be attending the SeQueL meeting (see July issue of Quanta for details) on 24th October 1998 to sell copies of my range and also to give a talk on the Pointer Environment. All are invited (beginners and techies alike).
For prices, see my advert in this issue.
... and some last minue news: QL genealogist, the much acclaimed pointer driven family tree program has now been re-released. This program allows you to store details on a family tree, including notes, text files and pictures connected with each person. Record relationships, important events and dates, and then the porgram can use this database to create a full family tree. Upgrades from earlier version are available, as well as Windows versions for the PC - ask for full details. Rich Mellor

Bill Richardson

Following the death of Felix Fontain his QL partner Bill Richardson has to dispose of QL and Spectrum products which have been stored at his house. He will be advertising various items such as 4MB floppy drives and assembled twin drives at half price (See his advert for details). Other items to dispose of are QLs and new circuit boards, microdrives for both QL and Spectrum. Lots of mains units, QL keyboards and ICs for both QL and Spectrums. Apart from products advertised he invites enquiries for bulk quantities if anyone is interested.

Qubbesoft

QL Emulator for Amiga has recently been upgrade to V3.24. This upgrade now includes access to hard disk for A600, A1200 and A4000 via QUBIDE

First Winner

Do you remember our request for Hot Tips? Here is a really good one! If you send in your hot tip, please make sure that it is short, easy to use and easy to remember - not long program listings!

This issues's winner is **Kit Lester**. The **voucher of £10** will be in the envelope with your QL Today, Kit! Here is his hot tip:

Using "_"

I never did quite master the use of omitted parameters in the TK2-and-successor wild commands - it all seems very quirky - but have stumbled on a useful undocumented feature that seems to work well - and even the same on TK2 & SMS!
Simply, a single underscore seems to stand for the current DATA_USE directory. Hence

```
WCOPY _ TO BLAH_
```

copies all files from the DATA_USE directory to BLAH_, and

```
WCOPY BLAH_ TO _
```

does the reverse.

Editor's note: The underscore in the first example can be omitted, and DATA_USE will still be used. However, if it is omitted in the second example the DEST_USE or SPL_USE setting will be used, which can result in TO par_blah. To avoid confusion, it is a very good idea to use it in both directions just as shown in the examples above! If you are confused now, try various settings for DATA_USE, DEST_USE and SPL_USE, try different WCOPY parameters and you will see the differences. You can always abort, so don't worry about copies all over your floppy/harddisk - or best play with your RAM-disk!

code. In the past many users of the emulator have been asking for access to their Hard Drives from the QL side and now they have it. QUBBESoft P/D have let the Authors of the Emulator use the QUBIDE code for this reason. This does not mean to say that the QUBIDE ROM or code is Public Domain as the source code for this is not included in the package. Other minor bug fixes have been included in the upgrade. The Emulator comes on three Disks, two of which are in Amiga format and the third being in QL format which includes lots of Utilities. Available now at 3 pounds plus 50p P&P.

I have just heard that within the past week SyQuest has decided to discontinue the SyQuest EZFlyer 230. This is very unfortunate as the EZFlyer has been a very popular product. SyQuest tell me the reason for this is because sales have dropped off and time moves on and more and more people are demanding larger capacity from their removable media. The alternative to the Flyer is a product called the Sparq, which is available in EIDE format but I haven't tried it out on QUBIDE yet. We are currently working on a new QUBIDE ROM that will introduce the ability to connect the Iomega ATAPI IDE Internal Zip drive to QUBIDE. We also hope that this upgrade will also cater for the LS120 ATAPI IDE drive, but we do not have one to test yet, so if anyone out there has one they can let us borrow we would be very grateful. Thats all for now folks.

Editor's comments: the discontinuation of SyQuest products is quite annoying: first the 105MB, then the 270MB, then down to 135MB, and again up to 230. A lot of incompatibility! Before you buy any more SyQuest products consider the following news: in a dealer magazine, I found the news that SyQuest gets rid of over 80% of the staff in the USA - and although the SparQ seems to be a well selling product, SyQuest shuts down one of their two factories in Fremont.

Still no reason for the competitor to smile: IOmega had to declare major losses (39 million $'s, if I'm not wrong). Even worse, IOmega bought Nomai, the third player in the removeable drive world (the Nomai 750 could handle 540 and SyQuest 270 media as well) - but it seems that after the takeover by IOmega no drives are made anymore. My suggestion: if you consider bying a removeable drive, wait until the situation clears up.

QL ROM Copyright

Amstrad PLC have clarified via a reply to Tony Firshman the situation regarding the distribution of QL ROM images. Cliff Lawson said "If Amstrad do own the rights to the QL ROM then just as for CPC and Spectrum ROMs we happily give emulator writers etc. the right to include a binary image of our copyrighted material as long as our (c)opyright notices remain unchanged and we appreciate it if the manual/ software includes a message to the effect of "Amstrad have kindly given their permission for the redistribution of their copyrighted material but the copyright remains with them."

QL News List

The QL News List is a free email based news service. It dervied from the email Update Survey which JMS started some months ago. Not much has happened in terms of replies, which shows that it is not worthwhile to introduce such the update service. However, there was some interest in a news service, so here it is! Everybody can join the list, it is absolutely free, and you can unsubscribe at any time. You can subscribe via the JMS website at:

www.j-m-s.com/smsq/qlnews.htm

or just send an email to

qlnews-request@md.gen.com

and insert "**subscribe**" (without the quotes) into the text field. You will then automatically get news about QL shows, new hardware and software developments etc. via email. Once again, this is a free service. Please join - stay informed!

Jochen Merz Software

There are plenty of new versions as you can see in my slightly redesigned ad.

A lot of software development is going on, as you will when you read this issue of QL Today. We hope to have the announced items ready soon, hopefully for the Austria QL show or the Eindhoven international meeting - at the moment things look pretty good.

I am going to prepare a batch of update sheets for the QDOS/ SMS Reference Manual, so if you have found any mistakes in there, please let me know.

TurboPascal for QDOS?

Simon N. Goodwin writes:

There's a new Turbo-Pascal compatible compiler just out as Amiga freeware. Do you know of anyone who'd like and be able to do a port of it to Qdos?

I got the original hint from a note posted to the Usenet group **comp.sys.amiga.programmer**

There's a lot of nice software available as TP source, if the compiler is available, and personally I'd consider it a better language than C for new developments which need to be portable.

The author (Carl) replied that although he didn't have the time to add full support for QDOS, he would add support for the QDOS platform in the compiler if someone else could do the rest of the work (presumably QDOS libraries etc).

Paragraph

More features in the next issue - too long to be published here, as it arrived after the deadline. ■

RWAP QL SOFTWARE

All software only available on 3.5" disk
Manuals all supplied in Quill format
All programs need 256K min. unless specified

WARGAMES

War In the East MKII v1.24 (Upgrade from original only)		£10
D-DAY MKII v3.03 (The Allies Take on the Germans)		£15
Grey Wolf v1.7 (Graphical Submarine Adventure)		£10

ADVENTURES

Return to Eden v3.08 (Graphics & Text Adventure - 3 disks)	NEW VERSION	£15
Nemesis MKII v2.01 (Text Adventure)		£10
The Prawn v2.01 (Spoof Text Adventure)	NEW VERSION	£10
Horrorday v3.1 (Humorous Text Adventure)	(128K)	£10
West v2.00 (Real-time Text Adventure)	NEW	£10
The Lost Kingdom of Zkul (Real-time Text Adventure)	NEW	£10
Adventure Package 1 (Nemesis MKII, The Prawn & Horrorday) only		£25
Adventure Package 2 (Return to Eden & any other adventure) only		£20
Adventure Package 3 (West & The Lost Kingdom of Zkul) only		£15

GAMES

Open Golf v5.19 (Golf Program - Good Graphics)	(384K)	£10
Quizmaster II Package (Original + Question Module 2)	(128K)	£7
Stone Raider II v2.00 (Boulderdash clone)	NEW	£5

UTILITIES / GENERAL INTEREST

Quick Mandelbrot III v3.01 (Explore the Mandelbrot Set)		£10
3D Terrain v1.23 (Displays Abacus Files as Pictures)	(196K)	£10
SToQL v1.29 (Converts ST screens to QL format)		£10
Q-Route v1.07 (PD Demo Version Route Finder Program inc map)	(796K)	£2
Q-Route Britain Map (As at 6/7/98 - needs Q-Route!)	NEW VERSION	£2
Flashback SE v2.02 (Upgrade from Original Only)	(128K)	£2
QL Genealogist v3.20 (create a family tree)		£25

SPECIAL OFFER

Order before 31st October 1998 and receive 20% off all orders over £5 of the above items).

NOW RELEASED

The SBASIC / SuperBASIC Reference Manual (this is the book everyone is talking about - 1000+ pages of essential information for the BASIC programmer, together with electronic index, example programs and public domain toolkits). The price of this book is **£40 plus post and packing** (£7 UK, £12 Eire, £22 Europe, £32 Far East & Australasia, £25 Rest of World). Why not pre-order and collect at a show near you??

HOW TO ORDER

CONTACT: Rich Mellor, 26 Ashenhurst Road, Russells Hall, Dudley, West Midlands DY1 2HH
Tel: 01384-350043

If your system can read HD or ED disks, please specify.

(Payment must be cheque in Sterling payable to R. Mellor)
(Credit Card Users can order via QBranch)
Visit our Web Page: http://www.qbranch.demon.co.uk/rwap.html
Send an S.A.E. and blank disk for our current catalogue

IBOX and how to make money

Stuart Honeyball

TF Services will soon be launching a new manifestation of the QL. It will have no keyboard, no display and no disk drives. It will, though, have a 68020, a multi-purpose I/O port, a network, 2 serial ports and will run Minerva. So what is it?

IBOX is aimed at the fast growing real time control market. You may have heard of PLCs (programmable logic controllers). These are used to control industrial processes but are in themselves fairly simple often having only a few hundred program steps. IBOX will have up to 2M bytes of RAM and can thus trump them on programmability. Also IBOX will be programmable in Superbasic and so development will be very quick. Another advantage of IBOX is that if the 20 I/Os it has are not enough then one can simply add another IBOX and network them.

The network/serial ports form the real key to expandability. It will use either 1 or both serial ports depending on the required implementation and will allow a number of IBOXes to communicate with each other.

Communication with the host will use 1 of the serial ports. The host will be a terminal or emulator and thus can be a QL, a PC, etc.. The terminal screen will appear to be a QL display with the 3 Superbasic windows and so anyone reading this is likely to be familiar with programming IBOX even before they've seen one. The development system, i.e. the Superbasic interpreter and editor, will actually be inside IBOX and so no special software need be loaded onto the host. Disks with development code will be read and written to via the host's disk drives using the download and upload terminal facilities. Once the program is developed or loaded the IBOX can be detached from the host and run in the target system. All 2M bytes of RAM are battery backed so this serves as backing store otherwise known as RAM disks.

There will be about 20 I/O lines on a 37 way 'D' connector which wil include analogue inputs for reading real world quantities e.g. temperature sensors, strain gauges; high current (0.5A) drivers for relays, small motors, lamps, etc.; and general purpose logic lines (inputs and outputs). The whole thing will be quite small and consume about 0.5A from a 7V to 13V supply.

The major turning point in the QL saga is that this is a product specifically designed to appeal to non-QLers. We need your help in finding customers for IBOX so that the QL technology can ride on an expanding rather than the currently diminishing market. The key advantage of IBOX is the ease and speed of program development using Superbasic combined with plenty of RAM and processing power. For instance, if you wish to build a machine that could benefit from computer control then IBOX can save the effort of building the control elecronics and can be programmed and tested while simultaneously connected to both the host and the target. Another advantage of having the serial ports is that a single IBOX or a network of IBOXes can be left connected to a host terminal so that commands and status can be written and read to and from the IBOXes during normal operation.

If you are interested in selling IBOXes on at a profit and want to make some money from the QL then please contact Tony Firshman at TF Services. You could contribute to reviving the flagging fortunes of our wonderful operating system and get it out to the wide world where it belongs.

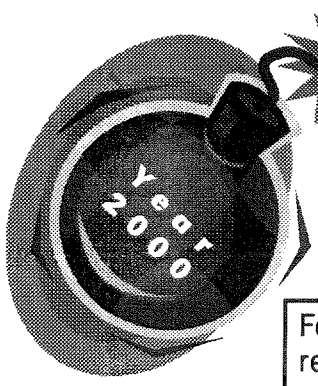
Good hunting!

IBOX - Intelligent peripheral controller

Preliminary spec:

- 68020 processor at 16 MHz
- Up to 2mbytes battery backed low current static ram (32 bit data bus to processor)
- Multitasking operating system and programs/data in RAM. Operating system based on Minerva
- Programming can be done in Basic, compiled Basic, C, or machine code.
- Two high speed RS232 serial ports for printing/networking.
- Serial terminal to ANY host computer for screen and programming.
- Operating system and multitasking programs uploaded from host computer. Can be developed and tested on an external emulator on many computers.
- Once IBOX set up, then host computer no longer needed.
- LED status indicators.
- Analogue/serial I/O for external control - relay drivers incorporated.
- Power and I/O via 37 way D connector. I²C I/O bus - can interface to existing TF Services products (see I²C interfaces) and other I²C interfaces.
- I/O and real time clock via PIC microcontroller which which provides some crash recovery capability - assuming static RAM contents are OK).
- On board power drivers for relays etc.
- Size - approx 80 x 69 x 14mm including connectors

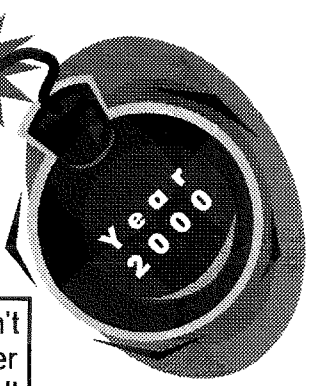
All these specs are design objectives, and are likely to change during development. Feedback at this stage would be very welcome.
TF Services.



Cover-Story

The Year 2000 Problem

Jochen Merz



For most QLers the year-2000 problem doesn't represent a problem - one changes the clock over manually in case the automatic change fails. It becomes more critical with QLers who use their computer in business.

A QL will change problem-free in the year 2000. QDOS stores the date in the "seconds" format, there are no special fields for minutes, seconds, hours, day, month, year, century. The "seconds" format is converted by different routines (in BASIC this is the function DATE$) into ASCII. As long as one does not further calculate with the converted data, nothing at all can actually occur. You determine the output format in most cases, e.g. with the TK2-clock:

```
CLOCK #1,"%d $m %y"
```

or

```
CLOCK #2,"%d $m %c%y"
```

You determine how it should look. Since also the file date and most other programs store everything as "seconds" long word, we definitely have no problem over the next few years (it gets very critical only in February 2097, but I guess we do not need to worry about this).

What happens if the computer is switched on and off? Where does the date come from? This depends mainly on which hardware you run your "QL". A genuine QL does not have a clock, unless a QIMI is installed - this is safe in both cases.

GoldCards and SuperGoldCards come with clock chips...

ATARIs also have inbuilt clock chips, the Mega ST has a different one than the Mega STE and the TT. With the mega ST the date of TOS changed correctly: from 31/12/99 to 01/01/00... and under SMSQ/E we get the correct date automatically, 01.01.2000.

It was shown in PC investigations that 93% of the BIOSes installed before 1997 and 47% of the BIOSes installed in 1997 cannot cope correctly with the year 2000! The actual problem is however the clock chip in the PC, the CMOS RTC. The investigation did not test the RTC, therefore 80 to 90% of all PCs might have problems with the year 2000.

When starting off a PC, there are actually two clocks running: the CMOS RTC and the system clock of the operating system. Upon switching on the BIOS calls the RTC and passes the date on to the operating system. Most RTC do not handle the century byte (19xx or 20xx) automatically when the century changes, therefore BIOS updates do not help much.

We can only guess what SMSQ/E makes of falsely supplied dates. Everybody is welcome to try it yourselves and tell us. Possible reactions of a PC are conversions to 1 January 1900 (QDOS, however, starts counting on 1 January 1961). With some PCS the date will reset itself back to 1984. That should resemble "only" to the same false date under SMSQ/E.

As already said the year-2000 problem is not a problem for most QLers. Data in the "seconds" format can safely be used for the calculation of date differences, so a subtraction of the two dates will always be correct. If data were stored however in the format dd/mm/yy, then it can cause problems - and most likely will!

The whole situation does not concern me personally too much: Since I use ATARI TT's under SMSQ/E as my main machines, no problems are to be expected here. My PC laptop could have difficulties, but it dates from 1997 so the chances are good that it will work.

Worst case: I have to adjust the SMSQ/E clock manually - also no major problem. My main PC is used mainly for emails and internet - the board also dates from 1997. I do not use the PC for invoicing, data bases (all done in SMSQ/E) and such things, so even if it is not 2000-proof I should not run into major trouble - I hope!

Before a simulation of the year 2000 (by changing the clock over) one should in any case do a backup, or better two! But shouldn't you have backups anyway?

■

"QL" or the Alternative?

Ian Pizer

By "QL" I mean any combination of hardware with QDOS or SMSQ software.

Years of QL then "QL" then AURORA meant I could mostly do what I wanted with my set up and, if not, there are many experts out there who are only too willing to assist.

In addition there is the continuous development of hardware and software to keep up interest and exercise the brain. There is QUANTA Magazine monthly and QLToday bi-monthly both bringing new ideas and information. In addition there are many friendly bulletin boards from which programs can be downloaded, information received, and if you are clever, upload your latest idea. To correspond with a bulletin board there are several programs available including the well known QTPI.

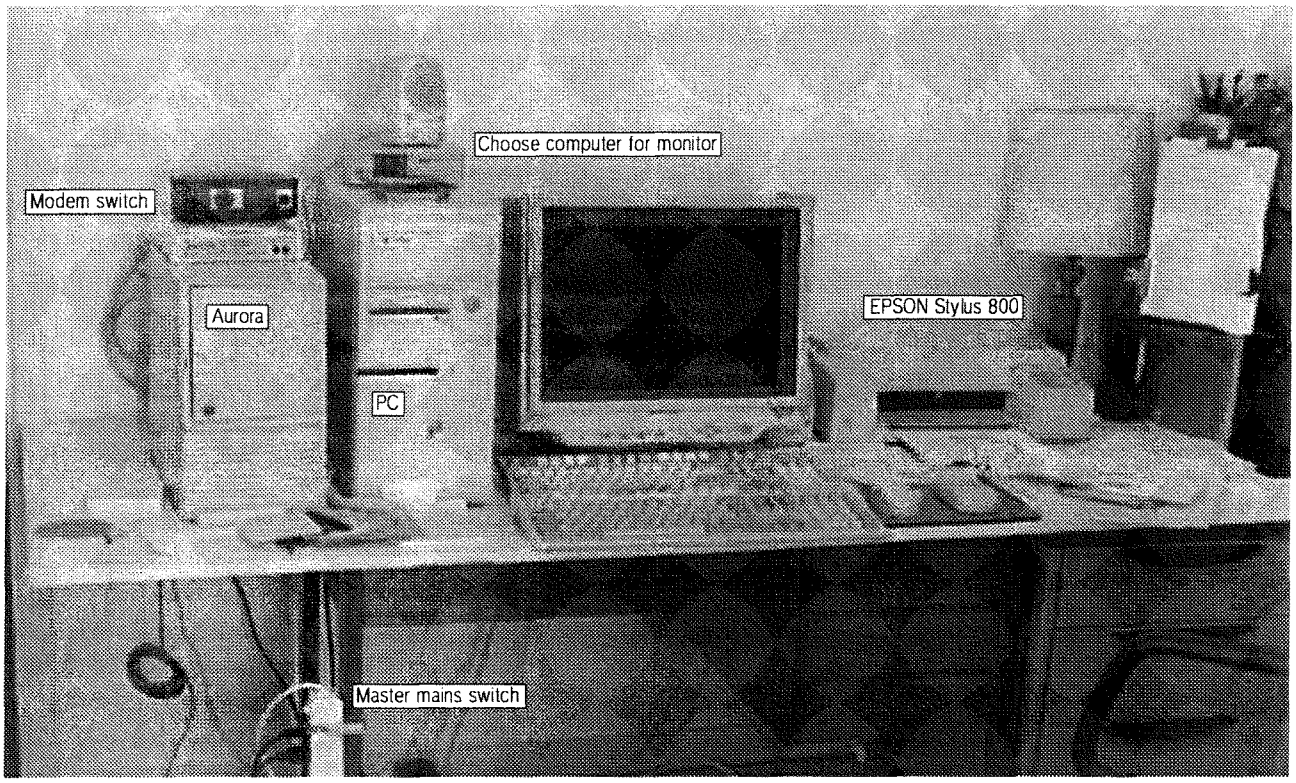
If you are more sophisticated you can send out a FAX. If you have the time or inclination you can visit a QL meeting and speak to the experts about problems, or the vendors (who are also experts), buy the latest goody, and get updates of your favourite programs.

In addition I was lucky to have the possibility to use a PC to visit WEB sites which specialise in QL software and general information. What is more, each WEB site usually has links to other "QL" sites to add to the possibilities.

Also, you can download programs onto a PC floppy and use it on your "QL" if you can read PC disks (SMSQ/E allows this). That is a pretty hefty positive situation for any computer. So why did I obtain my own PC and am I glad I did so?

Firstly I was frustrated by the lack of the possibility of using CD_ROMs on a "QL" as there seemed to be a large volume of CDs full of exciting information even the whole of the Encyclopaedia Britannica (if you can afford it you need it!). Then there was the possibility of "surfing" the WEB which consists of connecting to known interesting WEB sites or asking one of the many search engines to search for whatever you want. Both these possibilities would be available with Windows95 on a PC and absolutely simple to manipulate by even the least computer-oriented person (so you might hope!). And I could install an emulator later and use the PC hardware as a standby if my QL/AURORA failed.

So off to the shop (actually shops, to check the market) and soon I was home with the beast. I chose a vendor with a parking place next to his shop in case repairs or help was needed (it was, because a vital rear connector was not fixed to the rear panel). The vendor also spoke English (he is English) a wise choice here in the French speaking area of Switzerland. I had planned to use the same monitor already connected to AURORA so I installed a Switch Box for that.



Ditto for the Modem. For the printer I found an automatic switch which knows which computer wants to print (clever switch!). Each machine has a different keyboard connector so I decided not to enter into that area and just have 2 keyboards. This has the disadvantage that often I start with the wrong keyboard and wonder if something has died. Perhaps it is possible to share one keyboard. If yes, can someone tell me how?

So what about my PC? Naturally it has taken me many hours to get to grips with this new system. A certain knowledge of MS-DOS from using Conqueror has been useful though not essential, and vital help from friends.

I bought a recommended book "The Mother of All Windows 95 Books" by Leonhard & Simon, 900 pages + CD-Rom, as a safety reference when I get lost and frustrated. Yes, CD_ROMs work, even play music if you so wish. Yes, I have found my way around the WEB (World Wide Web) and have downloaded QL files (also some PC files). But I still use "QL" for e-mail, Bulletin Boards,programming in Basic, writing text in Text87 or QD, alarms, etc. I felt lost without the possibility of having ALT-KEYs so found a PC program that does that (DOSKEY).

W95 is infinitely complex. The user is no longer really in charge as one can sometimes so feel with "QL". If you load a new facility it loads a family of programs which go somewhere into the system and if you want to remove that facility then you do not just delete one file but use an uninstall program. One annoying feature is the shutdown procedure. You do not just switch off but hit 3 mouse buttons then the computer will shutdown for you which takes about 15 seconds, but I soon adapted to that. My "QL" takes 62 seconds to start, my PC takes 69 secs. Of course it is niceto see many colours on a colour monitor! It would take me many more hours to learn to do all the things I can do on "QL" so am I glad I bought the monster?

There will now be a break in transmission for a self-interest advertisement "look at **http://www.dvdesign.com** for Computer Art and interesting Poetry". (Will they = QLT, print an ad. like that?). *[**Yes, but the bill for the advert is in the post to you - Editor**]*

I think I must say yes, mainly because the WEB opens up many possibilities; the world is really out there waiting to get in contact with you, and because, given time, I might get near to feeling I am its master rather than its slave. I hope I can find enough contact with experts who can tutor me when necessary so I can feel comfortable as I do with the "QL" situation.

There are many features in W95 that I do not need and many that I may never master and many that I have not found. It is all a matter of time, patience and frustration.

However we must continue to shout - "QL" forever! It is after all a damn-good and expanding system. ■

Gee Graphics! (On the QL?) - part 6

Herb Schaaf

How far is the long way around the ellipse?

Well I thought this would be easy to look up, but found out again how little I know, and that there is so much more to learn.

My original idea was to take a circle and gradually flatten it via a series of ellipses into a line while maintaining a constant perimeter. Let's say we start with a circle having a diameter of 1 and thus a circumference or perimeter of PI = 3.14159...; as we flattened it we'd finally end with a line that looked to be PI/2 long with a perimeter of PI. How do we size the ellipses used to make the series look like a smooth transition? All we need is a way to figure the perimeter of an ellipse. In the 17th and 18th centuries this problem was referred to as the "rectification of the ellipse", and many wrestled with it. Eventually it seems to have led to elliptic integrals, elliptic functions, complex variables, and other fascinating branches of mathematics that I don't understand.

Being blessed by having access to the Morris Library at the University of Delaware I was able to find a few ways that had been used to approximate the perimeter of an ellipse along with other (new to me) concepts such as multiple factorials and the AGM or Arithmetic Geometric Mean.

The mathematical texts on elliptic integrals have a variety of conventions when describing the ellipse, some referring to the ellipticity e as a k factor they call the modulus or else as an angle ASIN(k) which they call the amplitude. In this way an ellipse with an eccentricity of .5 could be listed in the tables as k = .5 or as ASIN(k) = 30 degrees. In this system the circle would have a k factor of

0 and could be listed as 0 degrees, while the flattened line would have a k factor of 1 and could be listed as 90 degrees.

Elliptic integrals come in at least 3 classes or kinds. They have rather mundane names: elliptical integrals of the first kind, of the second kind, and you guessed it; of the third kind. The first kind, K(k,phi), is useful when working with simple pendulums that swing in wide circular arcs. The second kind, E(k,phi), is a measure of the arc length of an ellipse; just what I was looking for! I have no idea of what they use the third kind for; maybe to work out the Chandler wobble in the spin of the earth or Fermat's Theorem? The tables of the "complete" (when phi = PI/2 or 90 degrees) elliptic integrals only concern themselves with 1/4 (or quadrant) of the ellipse, going from the point on the ellipse at the minor axis along the ellipse to the point on the ellipse at the major axis. There are also more extensive tables for the "incomplete" elliptic integrals which give values for going only part of the way along the ellipse from the minor axis towards the major axis.

All the methods for calculating the elliptic integrals seem to use approximations found by summing up a series of terms, and if the terms eventually get quite small then we can stop whenever the error of neglecting the smaller terms would be less than the accuracy (or is it precision?) of our computer or our need or our ability to measure and/or display the results.

I found the history of the search for the solution to elliptic integrals diverting; what follows are the names of some of the seekers and when they did their work. Huygens in 1673 worked out what he needed to make a better clock that was less affected by how wide the pendulum might swing. Newton in 1676 was able to figure the perimeter when the eccentricity was not more than 2/3. Fagnano starting in 1714 added more insights, and John Landen put a transformation together in 1755 that provided a way to calculate the perimeter of ellipses. Legendre worked on the problem from 1786 to 1832, publishing 10 place tables in 1816. Gauss had the idea of the AGM in 1797, but did not publish it because the French Academy had "sneered" at his "Disquisitions" in 1800. Historians say Legendre could have been saved 26 years of calculations had he known of the idea of Gauss. Ivory developed a series in 1796 for more rapid calculation of the perimeter. Peano in 1887 used continued fractions to find an approximation for the perimeter of the ellipse using the second convergent; the same formula was published in 1889 by Boussinesq. Peano was then quick to point out his own priority and elaborate on the how small the error could be after the third convergent. Makes you wonder about the fourth and further convergents. In 1914 Ramanujan 'empirically' developed 3 quick approximations with extremely small error. In 1930 Goormaghtigh blended one approximation that had a positive error with a second approximation having a negative error, resulting in an expression that was said to be so good that it could express the distance the earth had traveled around the sun in a hundred billion years to within the thickness of a sheet of paper. In 1978 Nyvoll gave a short formula that is probably good enough for most of us. In 1987 Borwein showed the power of the AGM (Arithmetic Geometric Mean) for numerical analysis and calculations, including the elliptic integrals. I've written a QL DEFine FuNction AGM_ellipint using the AGM approach, and it is included in the listing, AGMellipse_bas. At extreme values when k gets close to 1 (or phi gets close to 90 degrees) things get tricky; push the limits to see if you get strange results!

Part one of the program "calc_n_show" asks for a perimeter, and then some other ellipse parameter; it will work out the remaining parameters for the ellipse and display the numerical results, pause, and then draw the ellipse. Part 2 of the program "same_perim" has the QL draw a series of ellipses from the circle to the line, all having the same perimeter. Part 3 of the program "same_area" has the QL draw a series of ellipses that all have the same area. This is rather straight-forward because there is a simple formula for the area of an ellipse. Area = PI * a * b, where a is the semi-major axis and b is the semi-minor axis. We stop before we reach the line of infinite length, but go far enough to see that we have begun to confuse those QL's that don't have SMSQ/E.

As you watch these ellipses collapse from the circle towards the line, notice that most of the 'action' takes place at the bottom or the top; it depends on which QL I run it on. It does make me wonder what marvelous math manipulations the QL is going through when it plots ELLIPSE.

Next time I hope to explore the incomplete elliptic integrals and find the arclengths for selected parts of the ellipse.

and here's the listing

```
100 REMark AGMellipse_bas
110 REMark H L Schaaf Aug 7, 1998
120 REMark to accompany Gee Graphics part 6 in QLToday
130 :
140 WTV : PAPER 0 : INK 7 : CLS : menu
150 :
160 DEFine PROCedure calc_n_show
170 REPeat queries
180 CLS  :  CLS#0
190 INPUT \,"Perimeter of Ellipse ? (more than zero, less than 1E308) ";p
200 CLS#0
210 IF p<=0 OR p > 1E308 THEN
220  PRINT#0;p;" out of range, use a positive value less than 1E308"
230  GO TO 190
240 END IF
250 SCALE p/2,-p/3,-p/4
260 CLS
270 PRINT\ "the perimeter is given as ";p
280 PRINT\ "e = Eccentricity can range from 0 to 1 (also called k)"
290 PRINT\ "d = equivalent angle from 0 to 90 degrees  { == ASIN(e) }"
300 radius = p/(2*PI)
310 PRINT\\ "a = semi-major axis can be from ";radius;" to ";p/4
320 PRINT\ "b = semi-minor axis can be from 0 to ";radius
330 INPUT\ "which one of these will be given; e OR d OR a OR b ?",ans$
340 IF ans$ == 'a' : find_be
350 IF ans$ == 'b' : find_ae
360 IF ans$ == 'd' : convert_d_to_e
370 got_e = 0
380 IF ans$ == 'e' : find_ab
390 IF NOT((ans$=='a')OR(ans$=='b')OR(ans$=='d')OR(ans$=='e')) : GO TO 260
400 CLS : CLS#0
410 :
420 give_answer
430 PRINT #0,,"touch [spacebar] for illustration"
440 PAUSE
450 CLS: CLS#0
460 draw_ellipse a,b
470 PRINT #0,,"touch [spacebar] for another ellipse, [ESC] for menu."
480 IF CODE(INKEY$(-1))=27 : CLS#0 : menu
490 END  REPeat queries
500 END DEFine  calc_n_show
510 :
520 DEFine PROCedure convert_d_to_e
530 INPUT\ "Degrees of inclination 'd' ? ",d
540 CLS #0
550 IF d<0 OR d>90 : PRINT #0;d;' out of range (0 to 90)' : GO TO 530
560 IF d >= 89.998 : d=90
570 e = SIN(RAD(d))
580 got_e = 1
590 find_ab
600 END DEFine convert_d_to_e
610 :
620 DEFine PROCedure find_ab
630 REMark given e, find a, and b
640 REMark e = (SQRT(a^2 - b^2))/a
650 IF NOT got_e : INPUT\ "Eccentricity 'e' ?", e
660 CLS#0
670 IF e <0 OR e>1 THEN
680  PRINT #0;e;" out of range (0 to 1)" : got_e = 0 : GO TO 650
690 END IF
700 REMark assume a = radius, calculate b, get p1 and compare with p
710 a = 1
720 b = SQRT((a*a) - (a*e)*(a*e))
730 d1 = DEG(ASIN(e))
740 p1 = PerimEllip(a,b)
750 REMark modify to fit
760 a = a * (p/p1)
770 b = SQRT((a*a) - (a*e)*(a*e))
780 p1 = PerimEllip(a,b)
```

```
790 END DEFine find_ab
800 :
810 DEFine PROCedure give_answer
820 CLS
830 INK 4
840 PRINT\ "eccentricity"; TO 14;"semi-major"; TO 26;"semi-minor";
850 PRINT TO 38;"perimeter" TO 50;"error"
860 PRINT e; TO 14; a; TO 26; b; TO 38; p1; TO 50; p-p1
870 PRINT\ "major axis = ";2*a,", minor axis = ";2*b
880 PRINT "F(k) AGM ( 1st complete elliptic) = ";
890 PRINT AGM_ellipint(SIN(RAD(d1)))
900 PRINT "E(k) AGM ( 2nd complete elliptic) = "; E_k
910 PRINT "major circum = ";2*a*PI,"minor circum = ";2*b*PI
920 PRINT "k = ASIN(e) = ";DEG(ASIN(e));"°"
930 INK 7
940 END DEFine give_answer
950 :
960 DEFine PROCedure find_be
970 REMark given a, find b and e
980 INPUT\ "Semi-major axis 'a' ?",a
990 CLS#0
1000 IF a < radius OR a > p/4 THEN
1010  PRINT#0;a;" out of range (";radius;" to ";p/4;")" : GO TO 980
1020 END IF
1030 REMark set b = a and then find p1
1040 b = a
1050 REMark adjust b until p == p1
1060 REPeat loop
1070 p1 = PerimEllip(a,b)
1080 IF p == p1 : EXIT loop
1090 b = b * (p/p1)
1100 END REPeat loop
1110 e = ecc(a,b)
1120 d1=DEG(ASIN(e))
1130 END DEFine find_be
1140 :
1150 DEFine PROCedure find_ae
1160 REMark given b, find a and e
1170 INPUT\ "Semi-minor axis 'b' ?",b
1180 CLS#0
1190 IF b < 0 OR b > radius THEN
1200  PRINT#0;b;" out of range (0 to ";radius;")"  : GO TO 1170
1210 END IF
1220 a = b
1230 REMark assume a = b (smaller circle) and then find p1
1240 REMark adjust a until p = p1
1250 REPeat loop
1260  p1 = PerimEllip(a,b)
1270  IF p == p1 : EXIT loop
1280  a = a * (p/p1)
1290 END REPeat loop
1300 e = ecc(a,b)
1310 d1=DEG(ASIN(e))
1320 END DEFine find_ae
1330 :
1340 DEFine FuNction PerimEllip(a,b)
1350 K_k = AGM_ellipint(ecc(a,b))
1360 RETurn 4*a*E_k
1370 END DEFine PerimEllip
1380 :
1390 DEFine FuNction ecc(a,b)
1400 RETurn (SQRT(a*a - b*b))/a
1410 END DEFine ecc
1420 :
1430 DEFine PROCedure draw_ellipse(a,b)
1440 LOCal i
1450 REMark red circle for major axis
1460 INK 2
1470 CIRCLE 0,0,a
1480 REMark green circle for minor axis
1490 INK 4
```

```
1500 CIRCLE 0,0,b
1510 REMark now the ellipse plotted at every degree in white
1520 INK 7
1530 FOR i = 0 TO 2*PI STEP PI/180
1540  x = a*COS(i)
1550  y = b*SIN(i)
1560  POINT x,y
1570 END FOR i
1580 END DEFine draw_ellipse
1590 :
1600 DEFine FuNction AGM_ellipint(k)
1610 LOCal i
1620 REMark Arithmetic geometric mean for elliptic integrals based on
1630 REMark Algorithm 1.2 in "Pi and the AGM" by Borwein and Borwein
1640 REMark     ISBN 0-471-83138-7   John Wiley 1987
1650 REMark solves F(k) = complete elliptic integral of first kind
1660 REMark   and  E(k) = complete elliptic integral of second kind
1670 REMark k is the eccentricity of the ellipse
1680 REMark k is often expressed in terms of alpha = ASIN(k)
1690 REMark k' is the complementary k
1700 IF k<0 OR k>1 : PRINT #0;k;' is out of range (0 to 1)':STOP
1710 k_comp = SQRT(1-k*k)
1720 REMark iter(1 for Arithmetic, Borwein's 'a'
1730 REMark iter(2 for Geometric,  Borwein's 'b'
1740 REMark iter(3 for Borwein's 'c', a measure of convergence
1750 DIM iter(3,1)
1760 iter(1,0) = 1
1770 iter(2,0) = k_comp
1780 iter(3,0) = k
1790 sum_of_terms = (2^-1)*(k*k)
1800 nth = 0
1810 REPeat converge
1820  iter(1,1) = (iter(1,0)+iter(2,0))/2
1830  iter(2,1) = SQRT((iter(1,0)*iter(2,0)))
1840  iter(3,1) = iter(1,0)-iter(1,1)
1850  sum_of_terms = sum_of_terms + 2^(nth)*(iter(3,1)*(iter(3,1)))
1860  nth = nth + 1
1870  FOR i = 1 TO 3
1880   iter(i,0)=iter(i,1)
1890  END FOR i
1900  IF ( iter(3,0) < 1E-10 ) : EXIT converge
1910 END REPeat converge
1920 REMark ratio of second kind to first kind
1930 sec_to_first = 1 - sum_of_terms
1940 K_k = PI/(2*iter(1,1))
1950 E_k = K_k * sec_to_first
1960 IF k = 1 THEN
1970  REMark K_k is actually infinity
1980  K_k = 9.9E615 : E_k = 1
1990 END IF
2000 RETurn K_k
2010 RETurn E_k
2020 END DEFine AGM_ellipint
2030 :
2040 DEFine PROCedure same_perim
2050 p  = 100
2060 SCALE p/2,-p/3,-p/4
2070 CLS
2080 FOR e = 0 TO .6 STEP .1 , .61 TO .95 STEP 1E-2
2090 got_e = 1
2100 find_ab
2110 CLS
2120 ELLIPSE 0, 0, b, a/b, 0
2130 END FOR e
2140 FOR e = .951 TO .99 STEP 1E-3,.991 TO 1 STEP 1E-4
2150 find_ab
2160 IF NOT(b) :  EXIT e
2170 IF b: CLS : ELLIPSE 0, 0, b, a/b, 0
2180 END FOR e
2190 CLS :LINE -p/4,0 TO p/4,0
2200 PRINT #0,,,,,"[spacebar] for menu"
```

```
2210 PAUSE
2220 menu
2230 END  DEFine same_perim
2240 :
2250 DEFine  PROCedure same_area
2260 REMark start with circle
2270 REMark equal area ellipses
2280 SCALE 4,-3,-2
2290 CLS
2300 prod = .25
2310 FOR b = .5 TO 9E-2 STEP -1E-3
2320 IF b : a = prod/b
2330 CLS
2340 IF b: ELLIPSE 0,0,b,a/b,0
2350 END FOR b
2360 PRINT #0,,,,,"[spacebar] for menu"
2370 PAUSE
2380 menu
2390 END  DEFine same_area
2400 :
2410 DEFine PROCedure  menu
2420 CLS :CLS#0
2430 PRINT\\"  This program can calculate complete elliptic integrals"
2440 PRINT,"of the first and second kind."
2450 PRINT\,'If given a perimeter and one of the following:'
2460 PRINT,'  ( major axis OR minor axis OR eccentricity )'
2470 PRINT,'it will return the other values and draw the ellipse.'
2480 PRINT\,'It can also show a series of ellipses from circle to line;'
2490 PRINT,'all having either the same perimeter, or the same area.'
2500 PRINT\'  Select by touching number key.'
2510 PRINT \,'[ 1 ]  Calculate Parameters and Show Ellipse'
2520 PRINT \,'[ 2 ]  Constant Perimeter Ellipses'
2530 PRINT \,'[ 3 ]  Constant Area Ellipses'
2540 PRINT \,'[ 4 ]  QUIT'
2550 a$=INKEY$(-1)
2560 choice = a$
2570 SELect ON choice
2580  = 1 : calc_n_show
2590  = 2 : same_perim
2600  = 3 : same_area
2610  = 4 : CLS : STOP
2620  = REMAINDER  : menu
2630 END SELect
2640 END  DEFine menu
2650 :
2660 REMark end of listing for AGMellipse_bas
```

Software Review

Thesaurus Review

James Hunkins

Jim, known for his extremely well done, fair and detailed reviews of QL soft- and hardware will not only review QL-Thesaurus in this article but also tell you more about the background and usage of this program.

Introduction

Geoff Wicks' QL-Thesaurus program has been around for some time and has been previously reviewed both in its original form and more recently in its pointer environment version. This article, while still being basically a review, will go into more depth and look at the actual usage of this particular program.

Before I started this article I thought that I understood what a thesaurus was. However, a thesaurus involves more than I had realized. According to the Funk and Wagnall's Standard College Dictionary, a thesaurus is 'A book containing a store of words, specially of synonyms and antonyms arranged in categories.' The key word here is 'categories.'

For years I have been using the Reader's Digest book, "Family Word Finder," and more recently the built-in thesaurus capabilities of several PC based word processors. It turns out that none were full fledged thesauruses. Each had both synonyms and antonyms. But they were missing the category part

of the definition. They were also somewhat limited in choices. At the end of this article, I will do a sample comparison so that this will become clearer.

In the QL-Thesaurus manual, Geoff Wicks corrects my understanding of what a thesaurus is with a proper definition. With the QL-Thesaurus program, he provides us with the real thing.

All this might seem a bit trivial, at least until you use QL-Thesaurus. However, have you ever been stumped for a word to use or perhaps been looking for an alternate phrase? Or how about searching for another direction to take your writing? Let's take an example from the manual. Looking up the word poisonous, the word can be found in two groups (more about this later); Human Life - Condition - Unhealthy, and Human Life - Danger - Danger. Let's take the second group where we can find items like 'jeopardy', 'in the lion's den', 'at the last extremity', 'dicey', and many, many more. One of the choices that caught my eye was 'sitting duck'. Now, if I was trying to write a story, this might have sent me along an entire new train of thought.

Basically, not only is QL-Thesaurus an extensive 'synonym and antonym dictionary', but it is also a wonderful source of what I would call 'thought triggers'.

The Program

The original program, which is included in case you don't use the pointer environment, runs as a standard QL program. It has all the basic options accessible as key presses and can be multitasked with your favorite word processor.

The newest version of QL-Thesaurus however is designed for use within the pointer environment and does the job very well. You must have the pointer environment to run this version. If you don't already have the pointer environment, it can be obtained from most bulletin boards and dealers. If you are using SMSQ/E, you already have the pointer environment built in.

In using the pointer environment, the program gains several advantages. You can use the mouse to quickly make selections, to choose multiple groups to look at, and to even move your window around. This last part is very useful if you have the higher definition screens and want to look both at your word processor and the thesaurus at the same time.

Geoff also did everyone a big favor with his implementation. He allows you to use either the mouse and/or single key strokes to make choices. Personally, I hate it when I use the mouse to make a selection on one side of the screen and then have to move my cursor all the way to the other side to make another selection. This seems to always occur no matter how carefully the user interface is designed. In QL-Thesaurus, all group and word selections can be made by mouse or cursor key control. All individual control selections are mouse or single key (underlined) selectable. You can select something with your mouse and then immediately, if more convenient, use a single key entry to make another selection without having to move the cursor.

Also properly done is a sleep choice (Zzz) which reduces the program to a button (QPAC II is not needed but if used will place the button into a chosen area on your screen, i.e. less clutter). The program also grays out selections that are not currently available.

Let's start using it

To look up a word or phrase, you have three choices; Thesaurus, Word, or Group. The most common method will be Thesaurus. With this choice you are prompted to enter the first letters of the word. It is best to only use three or four letters as, unlike a dictionary, every form of every word is not included.

After you enter the first letters, you get a list of words starting with the letters you just entered. You can page up or down through this list. This is actually the complete list of all words in the database. If you had patience you could page all the way to the top or bottom of the list (I would not recommend that unless you have nothing to do for quite some time). From the section of the list shown (or paged to) you can choose any word. The program will then give you either a list of different groups that the word appears in (a complex search) or, if the word only occurs in one group, the final list of words and phrases (a simple search)

See Figure 1 on the following page.

The Word entry option is very versatile but may overwhelm you with choices. If you are thinking of a word or phrase but are not sure how it starts, you can enter any sequence of letters that would be found in it. The Word option will return all words and phrases with those letters. For example, if you enter 'app', you get the standard list of words and phases that start with app such as appall, apparatchik (I have no

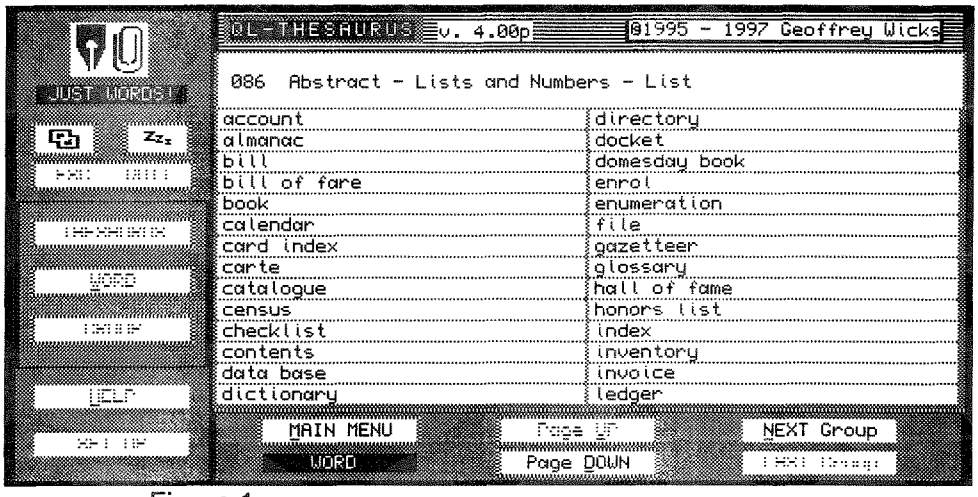


Figure 1

idea what it means either), apparatus, apple pie order, etc. You will also get words and phrases which include app somewhere within them, such as dapper, disappearance, bound to happen, cash-strapped.

Just the three letters 'app' returned over four full pages of selections. Again, this can be overwhelming. But it can be useful if you can't remember something exactly. It can actually be a bit fun to fool around with too. You never know what you will find. Hmmm, maybe I should go look up the word apparatchik... As one of the options for 'app' gave; 'wrapped in thought'.

The third option is Group. Words in QL-Thesaurus are found in 1000 different groups which follow Roget's Thesaurus groupings

These groups were originally published in 1852. From what Geoff says in the manual, most modern day thesauruses still use this original work as their foundation. I guess a well-done design never completely goes away (sound familiar?).

In the previous two search methods, whenever an entry had more than one group available, the groups were listed with their group number. For simple searches, the page displaying the search results contains the group description and number. The Group option allows you to enter the group number directly. You can also find a listing of the different groups in the 'groups.doc' file which comes with the program.

In the previous complex searches, you could go directly to the group by choosing the group name. The reason to have a separate group option, in addition to just exploring and looking for ideas, is to look up similar topics or antonyms. The groups are numbered in a very logic order. For example, look at the following six groups:

417 The Senses - Sound - Musical Instrument
418 The Senses - Sound - Hearing
419 The Senses - Sound - Deafness
420 The Senses - Light - Light
421 The Senses - Light - Dark
422 The Senses - Light - Dim

Group 418 is directly associated with 417 but has a different sub-category (Hearing vs. Musical Instrument). Group 419 would contain antonyms to group 418 (Deafness vs. Hearing). Continuing with the Senses, group 420 changes from Sound to Light. Group 421 contain antonyms to 420 (Dark vs. Light). Group 422 contains less extreme versions of the words in group 421 (Dim vs. Dark).

As you can see, by looking at the groupings you can find antonyms, similar words with slightly different meanings, and of course synonyms. It must have taken quite a bit of work to generate these groupings. My hat is off to Roget.

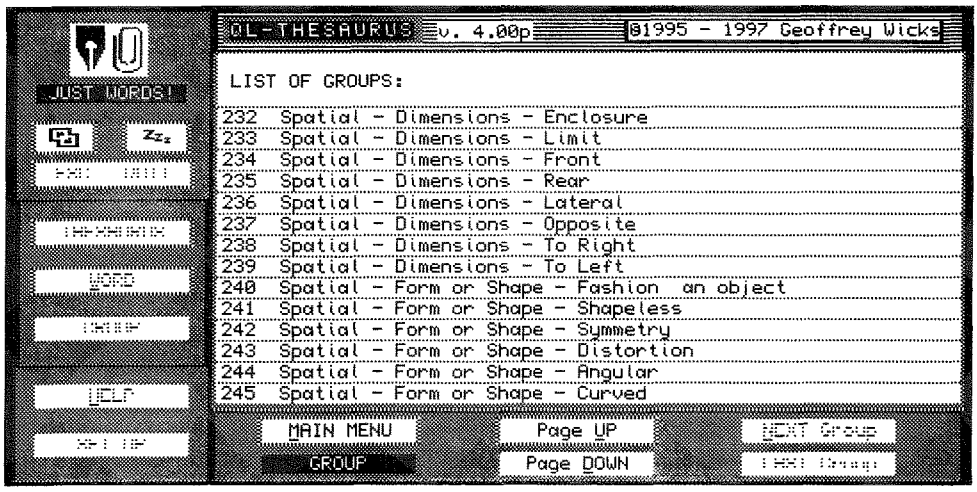


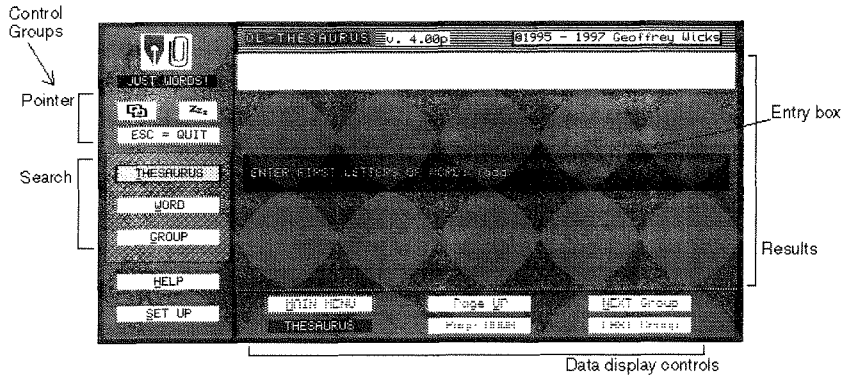
Figure 2

Search Types: The manual discusses two types of searches, complex and simple. Many words can be found in multiple groups. This can result from different uses of the word such as when it is used as a noun or as a verb. Many words also have multiple meanings (don't you just love English!). When

this happens, the program displays a choice of all the groups that your word is found in. You can then choose the group that you wish to view. If you would like to choose more than one group, using the space bar or left mouse button (LMB), highlight the individual groups that you want to view. Then by pressing the Enter key or right mouse button (RMB), you will be presented the first group of words. The menu options will let you scroll up or down through this group or change between the groups.
For the simple search, since there is only one group found, only that single group is displayed.
So that is it for using the program. It is fairly easy to use while returning a wealth of information.

Screen Layout

The screen is laid out cleanly.



The top level commands are on the left. They include the pointer options Move and Sleep plus standard options Quit, Help and Setup. They also include the three search options, Thesaurus, Word, and Group. Again, all non-selectable options are grayed out. The single letter keys to choose them with (versus pointer selection) are underlined.
The bottom of the screen has the data manipulation options. It lists which type of search is/has been done plus the choice to go back to the main menu. It also includes the page up/down and group movement selections.
Two thirds of the screen (middle right section) is dedicated to data input and display. All search information is input from this area.
The data for Thesaurus is displayed in a single column in this area while the Word results are displayed in two columns. The group data is displayed single column wide (each line does fill up the space) while the final search group contents (the words and phrases) are displayed in two columns.
Whenever a group of words is displayed, the top of the data area includes the group number and description, which stays displayed even when scrolling the group.

Manual / Help

The manual is, I am happy to say, fairly complete and simple to follow. It has all the basic sections; an introduction, system requirements, first essentials, quick start, booting up, program usage, etc. The author even includes background on the thesaurus and its origination, plus information on the database in case you want to modify it.
I appreciated the author including the quick start section for those of us who hate to read, plus screen dumps and even a simple index.
The on-screen help is concise (only 5 screens worth) and allows an exit to the main menu at any point. The only negative is that, while you can go forward, you can't go backwards through the help. A very minor point.

Setup

The setup was the only confusing part for me with this program. I run the program from my hard drive (floppy users should never have a problem) which I recommend because the database is large and takes a while to load from a floppy.
QL-Thesaurus uses two separate configurations. It uses an external data file where it keeps user configurable information such as the printer device, baud rate, etc. The location of this file is held within the program in the standard QL (QJump format) config space. The default location for this is the floppy drive.
Ah, there is the problem (unless you read the manual of course). If you copy the program to hard disk and then try to run it, the program keeps trying to look at the floppy for the data files (both configure and data base). If you have the original program floppy still in the drive all will seem well except that it will take forever to start (the data base is over 400K in size). If you have removed the floppy the program will tell you that it could not find the configuration file and abort.
The manual of course gives the clues needed to solve this, but does not come right out and make it clear what is hap-

pening. To run the program from a location other than flp1_ you either need to start the program with a parameter saying where the files are ‹lE: EX win1_thesaurus;"win1_"› or you can use a configure program to set the location within the QL-Thesaurus (such as CONFIG or MENUCONFIG). If you do either one, the program finds the files in the correct location with no problem and everything is wonderful.

I am guessing that the separate file with data is a left over from the original program version. I would highly recommend that all configuration information be put into the main program just as the file location currently is. Not only would this be simpler, but it would allow the user to update the file location (still used for the database) along with the other information either within the program or by the external config program capability.

The other configurable items are straight forward. ‹Sound› turns a beep response on or off, ‹Printer› redirects the final screen results to a printer, and ‹Column› toggles between 1 and 3 column output (this only affects the printer output, not the screen display). ‹Baud Rate› sets the speed if the printer is a serial device, and ‹Output› allows you to select the printer device (this can even be a file name if you want to save the results to a file for importation into another program).

These items are all configurable within the program. Setting them is done by clicking on the option and the possible values toggle on the screen for you; very clean. The only one you actually have to type in is the output device.

When you are done setting them, you can save the modified configuration items to disk, making them the default values every time you run the program. If you only want the changes for the current session of QL-Thesaurus, just click on ‹Return› and they won't be saved to disk. One reminder here, relating to the previously mentioned confusion that I experienced, is that while you can change the name of the device that you save the defaults to, this only changes where the file is saved to. It does not affect where the program will load the file from. Again, you can only change this location with a separate config program or as a parameter on the command line.

Tips

Here are a few tips that should improve your usage of QL-Thesaurus.

1. By selecting a word entry from the word list (either by clicking on it or by ‹Space› or ‹Enter› keys, the word is placed in a stuffer buffer. It can then be placed by the ‹Alt Enter› keystrokes back into either QL-Thesaurus or into another program.

2. QL-Thesaurus has a very large database and requires about 517K of memory to run. Obviously you won't be running it on an originally equipped QL without additional memory. With this memory requirement in mind, two items should be noted. One, loading it from floppy will take a while. So if you have a hard drive, you definitely should load it from there. If you are using the original Quill word processor that loved to grab all the memory it could find, you should load QL-Thesaurus before loading Quill.

3. I must admit, I did not read the manual at first. When I finally did I only read the 'First Essentials' section. I finally went back and read the full thing, finding that the manual was short and concise but full of simple and useful tips. It also contains information that made using the program simpler, faster, and most important, actually kind of fun. This is one of the few manuals that I have found that was actually useful for myself to read all the way through.

4. Geoff gives a very important recommendation on how to use this program. Don't examine in detail all the words and phrases that are shown. Instead, do a quick scan through them and wait for one to jump up at you. There are often a very large number of choices, many of which won't fit into what you are doing at all. While you could spend your time going through the results in detail, this will most likely make your writing 'artificial' and stilted. If you just scan, a word or phrase that fits into your writing style and theme will most likely be obvious. And much more pleasant to the reader.

Problems/Bugs

Before I start this section, let me state that the program has no problem that prevents it from being very useful and pleasant to use. The version that I tested is 4.00p and probably has been updated since then. Some of the few items that I found may have been fixed since I received my copy. However, just in case here are the ones that I did find.

When entering the location for the configuration and database files, there is a string length limit of 16 characters including the final '_'. If you enter a longer name as a command parameter, the program will not be able to find the files. If you enter a longer name from a configura-

tion program, it will prevent QL-Thesaurus from running and you will have to delete it and reinstall a fresh copy (always keep a backup, right?). The simple solution is to make sure you don't put the program and its files in a directory that takes up over 16 characters (including the device name).

Another problem occurs if you, during a Thesaurus complex search, click with the right mouse button outside the list of categories (such as in the bottom item selection area). This results in a 'QLIB Error 35, index out of range' error. Since you normally would not do this anyway, it is easy to avoid (yes, I test out illogical user input. After all, not all users are logical all the time, including myself).

I mentioned before the stuffer buffer that allows you to easily cut and paste words into other programs. Unfortunately in my case, I was only able to get it to paste back into QL-Thesaurus. Deadline limits kept me from investigating this farther. It might actually turn out to be a system problem that only I have. But since the function is so useful and in case others have similar problems it is worth mentioning (the only way to get something fixed is to let someone know).

My final comment in this section does not relate to a bug or real problem. I simply mention it to point out that Geoff has done some modifications to the original Roget thesaurus groups that his program is based on. Geoff mentions in the manual that he attempted to modernize Roget's terminology (? it is only 140 years old ?). He has also 'narrowed the ideas conveyed in many of the groups.' Some words were removed as they may have different meanings today which could be sexist, racialist or anti-Semitic undertones.

This is not a criticism. Definitely, in 140 years English has changed and I applaud Geoff's efforts to update the thesaurus. I bring this up to point out, as with any reference book or program, that some items may have been left out or interpreted differently from how you would view them. As it stands, QL-Thesaurus is a large and very useful thesaurus and stands well as a useful reference.

Recommended Improvements/Changes

No program is ever perfect or complete, especially when viewed by someone other than the author. Here are a few suggestions that, in my opinion, would be welcome improvements and enhancements to an already useful program.

I mentioned before that all the configuration information should be removed from the external configuration file and added into the internal QJump format config block.

In addition, while the program does a good job using the pointer environment, a resize screen would be useful. I would appreciate at times having a smaller area of the screen in use by QL-Thesaurus so that it would not be overlapped by my word processor which I prefer to dedicate most of my screen to. The current preset size is good for showing a decent quantity of results. Since it is easy to switch between QL-Thesaurus and another program while in the pointer environment, this is a relatively low priority suggestion.

All of the displays take full advantage of the display space within the program except for the Thesaurus search that only displays one column. It would be nice if the thesaurus search could also display two columns wide to show more results, as found in the other search screens.

Another nice modification would be to compress the database so that it does not take up as much memory when loaded. This might be made an option so that the compression would not be forced onto users with original QL speeds, but could still be used by faster QL compatible machines.

My last suggestion (a trivial one at that) is to incorporate and update the program's pointer environment additions into a single manual instead of having an insert.

Now, a Bit of Fun

A Comparison: here is a short comparison of some results from looking up the word 'hero'. The first results came from the PC world's Word word processor's built-in thesaurus. The second group of results were generated by QL-Thesaurus. Each is an abbreviated selection of the actual results. The QL-Thesaurus results shown here leave out the majority of the returned words/phrases due to space limits. Please also note that the method of displaying the results in this article in no way represents the actual program displays. Both programs do a good job of presenting the search results.

You should note that the Word based thesaurus is definitely a synonym/antonym lookup reference and could be considered more concise. The QL-Thesaurus on the other hand, while it can be used strictly as a synonym/antonym lookup reference, includes many related topics

Professional & Graphical Software

ProWesS

ProWesS is a new user environment for the QL. ProWesS is short for "PROGS Window Manager", but it is much more than that. Apart from a new window manager, it contains all the system extensions from PROGS, and is essential if you want to run programs which need these extensions.
The ProWesS reader is a major part of the package. It is a hypertext document browser. This means that text files which include formatting commands (including pictures) and possibly links to other files can be displayed and read in this program. This is used in ProWesS to read (and possibly print) the manuals, and display the help files. The hypertext documents which are used by the ProWesS reader are in HTML format, the format which is popular on Internet to display World Wide Web pages.
Another important aspect of ProWesS is the possibility to allow programs to automatically install themselves on your system, and to be able to run them without resetting the system. This means that, when you get a new program, all you have to do is insert the disk and indicate "start the program in flp1_", a menu option in the "utilities" button. To install a program, you indicate "install software", and the software can be added to your system. This way, you don't need to know how to write a boot file to use the multi-tasking capabilities of your computer.
ProWesS includes many programming libraries. These include syslib, an interface to the operating system, PROforma, a vector graphics system, allowing rendering both on screen and on paper (via a printer driver). The DATAdesign engine is also part of ProWesS. It is a relational database system with a bonus, as you don't even need a key field. You get a powerful record at a time data manipulation extension to the language you already use. Of course it also includes ProWesS itself, the new resolution independent window manager.

PFlist

Easy to use program to create listings on any printer (especially inkjet and laser). This ProWesS application allows you to indicate the files which have to be printed. Each column contains a footer which can include the filename and filedate. The listings always allow perforation. PFlist can create your listings in two columns and in landscape (or both).

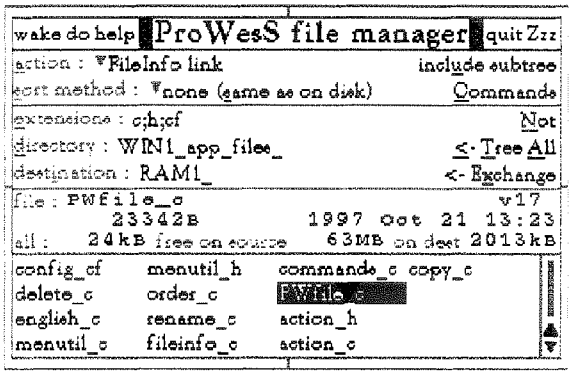
fsearch

File search utility with many useful options, like the choice to search only files with a certain extension, and whether or not the directory tree has to be scanned. All occurences of the searchstring will be displayed with line number or offset. You can also use special matching features, like case dependent, matching a space with a stretch of whitespace, and searching for a word dilimited string.

font-utils

manage your font collection. You can preview fonts on screen, see what characters exist in a font and convert Adobe Type 1 and similar fonts for use in ProWesS.

New ProWesS application
a powerful and very user friendly file manager



LINEdesign

Create artistic drawings, technical drawings, process bitmaps (even scale and rotate them!), and any kind of vector drawings. You can use grpahics objects to create the most fabulous drawings ever seen. Because LINEdesign is a vector drawing program, any part of the picture can be moved, scaled, rotated, slanted without any loss of precision or resolution. In LINEdesign, pictures are device independant, meaning that the printout will be the same on any printer (e.g. same size and position).
LINEdesign is good at handling text. You can easily put titles and full paragraphs on the page. All the fonts can be displayed at any size, rotation, etc. All the fonts which are available to ProWesS can be used in LINEdesign.
LINEdesign is a drawing program, but it can also be used by people who are not good at drawing. LINEdesign is a great program for making leaflets, posters, and any kind of printed work. Lots of clipart and extra fonts are available from public domain libraries and BBS's. You can even import Adobe Illustrator files.

DATAdesign

Never before has it been so easy to create, fill in and maintain your personal databases. To start a new file, just type the names of the fields. To add or delete a field, no problem, just do it. To change the name of a field, just indicate it. You can choose which fields are displayed and also which records. You can have a hidden comment for each record, look at the file in tabulated form and transfer data to the scrap or hotkey buffer. Files can be memory based (for speed) or disk based (for safety).

new address !!

Dr. Fr. Hemerijckxlaan 13 /1
2650 Edegem
Belgium

tel : +32 (0)3/ 457 84 88
fax : +32 (0)3/ 458 62 07
email : joachim@club.innet.be
www : http://www.club.innet.be/~year2827

ProWesS - BEF 2400 | DATAdesign - BEF 1200 | LINEdesign - BEF 1200 | PWfile - BEF 900 | fontutils - BEF 1200 | PFlist - BEF 600 | fsearch - BEF 600

Payment terms :

You have to run ProWesS to make LINEdesign, DATAdesign, fsearch, fontutils and PFlist work (even though DATAdesign uses wman).
All our software is normally supplied on high density (HD) disks. However they can be obtained on double density (DD) disks at an extra costs of BEF 100. To use ProWesS and any of our other packages, you need a system with at least 2MB of memory. You should have a harddisk although a two disk system will also work. The use of SMSQ/E is strongly recommended for optimal use of ProWesS.
If you are VAT registered (specify registration number) or live outside the EEC, the amount to be paid is the total (including postage) divided by 1.21 (no need to pay too much).
Payment can be done by EuroCheque in BEF, or by VISA, EuroCard or MasterCard. Credit card orders can be handled by phone. For credit card, please specify name of card owner, card number and expiry date.

Postage : Costs of postage and packaging have to be added. You can choose the quality. Rate depends on no of programs.

copies	*priority mail* Belgium	Europe	World	*ordinary mail* Belgium	Europe	World
one	100	200	240	100	120	145
two	135	340	420	135	190	230
3 or 4	160	560	770	160	310	395
5 to 8	185	870	1250	185	550	705
more	295	1130	1610	295	800	1030

All prices are in BEF, including 21% VAT

and phrases and is more likely to encourage new trains of thought.

Hero: Word thesaurus
- brave man (noun)
  - man of distinguished valor
  - champion
  - model
  - matyr
  - ace
  - winner
- male lead (noun)
  - protagonist
  - leading man
  - principal male
  - good guy
  - movie star

Hero: QL-Thesaurus
- hero
  - Feelings - Attitudes - Courage
    - achievement
    - adventurous
    - amazon
    - audacity
    - dauntless
    - defy danger
    - despise danger
    - dogged
  - Feelings - Status - Repute
    - accredit
    - aggrandize
    - apotheosis
    - celebrity
    - character
    - conspicuous
  - Moral - Good and Bad - Good person
- hero worship
  - Moral - Good and Bad - Approval
  - Moral - Religious - Idolatry
    - apotheosis
    - bow down
    - deify
    - sacrifice
    - satanism
    - self immolation
- heroic
- heroics
- heroin
- heroine

Don't You DARE! of course, as with any good thing, it isn't that hard to get carried away with using it. Here is a paragraph from elsewhere in this article, before and after I got carried away with QL-Thesaurus.

Original Text: In QL-Thesaurus, Geoff Wicks corrects my understanding of what a thesaurus is with a proper definition in the manual. With QL-Thesaurus, he provides us with the real thing.

Over Thesaurized (sorry) Text: In QL-Thesaurus, Geoff Wicks builds anew my understanding of what a thesaurus is with a befitting exposition in the handbook. With QL-Thesaurus, he delivers the authentic thing.

While the second version isn't too bad, can you imagine a whole article with this degree of word/phrase replacement? The point made is that any tool is only as good as it's use allows it to be. Please, use QL-Thesaurus and find your writing improving and probably being enjoyable. But don't get too over enthusiastic.

Finally...

It should be obvious by now; I recommend QL-Thesaurus. If you are using the original version, by all means upgrade to take advantage of a quicker and cleaner interface. If you don't have QL-Thesaurus yet, you should consider it a good buy. While QL-Thesaurus may not be useful for all your writing (it doesn't contain every modern phrase, including many found in technical writing) it does have enough content to allow it to smooth out even the dullest of writing subject matter.

Enjoy.

■

My christmas wish list for 1998

Jérôme Grimbert

Well, this is written in May, but it's never too early to ask. So here's the list of things that I would find great if they were available before the end of this year.

- support for CD-ROM in SMSQ/E of QXL2. (just like QPC, not really needed, but as it is possible... And it may be the killer gadget! (the CD-player program already exists, just missing the low level connection...)
- support for greater resolution in SMSQE of QXL2. (1024x768 and 1240x1024, just like QPC, again).
- quicker I/O (especially FLOPPY) for the QXL2 (well, I had the time to format and fill three HD floppies with my SGC, while only copying one HD floppy to the QXL2 ramdisk...) I know the QXL2 has to work with every intel (nobody can get an 80286 today), but at least a quicker solution based on a minimal 486 requirement should be available, even as an extra. (By the way, as it is MY wish list, the minimal requirement could go upto a 120 MHz (x2, bus at 60 MHz) Cyrix M1 6x86 (usually sold as a P150+).
- a working QL network for a full speed (25MHz) QXL2. Having to replace the standard crystal with a 20MHz is not a decent solution (well, why buy a Ferrari if you have to change the engine for the one of the Smart? I want the full Ferrari!).: Given the time for reading a floppy, transferring data between my SGC and QXL take hours of manipulation (and a lot of floppies).
- support for 16 colours, by using the unused flash bit. The eight new colours can be fixed for the whole system, or use the same trick as QPC. I would rather have a system width defini-

tion, as long as one can choose (configure) the palette [see my other article in the previous issue on the 16 colours, we are only beginning to evaluate what the best eight new colours will be!] [the palette should be specified using a full 24 bits RGB values, even if some PC hardware has to shift them to keep only 18 bits. This eases the testing/discussion phase, because it so easy to make an HTML file to display the possible colours, and HTML uses a 24 bit notation. The main points being that you can see the proposed colours, whether you use a Windows PC, a Linux PC, a Mac, or any X display, as long as you have access to a Web Browser]

For my part, I'm going to provide some new versions of my free software (pente, goban, chess[red queen], atome, triangle, carre, isola, rogue). The improvements will be, in this order:

- systematic support of 800x600 display, but still working on 512x256 mode 4.
- support of dual mode: mode 4 and mode 8
- extended mode 8, using QPC extension, for 16 colours. (that will be my way to promote the 16 colours, and my choice of colours). It will also work with UQLX, even in the 256x256 mode (Not everyone has a Minerva ROM to support extended resolution!)

I will probably have to write a decent sprite editor first (also freeware), or may be I will go with a simple text editor... I do not know yet.

Once done, I have some other game ideas, still board-kind, do not expect a doom-like from me! And if you have any game idea you want to share, just do it.

Santa (Jochen) replies: All the things you ask for are in the pipeline - in fact, more than 16 colours will be supported too. However, some things take longer than expected because other tasks (which earn money) change the order of priority sometimes. A lot of work has been done on a new, much faster QXL interface. Same is true for the colour drivers - from what I understand the main work is already done.

Nice to see that you're not only asking for things but you're also prepared to do something for it! If everybody would think (and act) the way you do, we would be a big step further on. However, one things should be clear: QPC accesses the PC hardware directly, the QXL needs to "talk" to the PC - this communication takes time and this means that the QXL will most likely not be able to reach the performance of QPC.

■

QLATter 1.109

by Al Feng

QLATter is a freeware utility intended for use with Jan Venema's QLAY emulator; but, it can be used with a "regular" QL or any other QDOS compatible, too.

The source code for the program has been corrected and updated so that the COPY function works, and the FORMAT function has been replaced with a M·K_DIR (faux/mock sub-DIRectory) option.

At the present time, HardCOPY remains non-functional due to limitations in the QLAY emulation; but, it will work on other QDOS compatibles.

QLATter supports easy sub-DIRectory access and is Minerva and SMSQ compatible. TK2_EXTensions are not required.

SELECT_DEVICE '0'

While QLATter's 'SELECT_DEVICE' option does not have "mdv()_" as a ready option, microdrives users can access the two devices on their QL via 'other'.

If you select 'other', then you can simply press 'm' and then the ‹ENTER› key, followed by either '1' or '2' and then the ‹ENTER› key, again.

Otherwise, simply move the green bar up or down using either the up-arrow or down-arrow key, or by pressing the first letter of the device name.

Change the device number by using the left_arrow or right_arrow key or by pressing a numeric key whose value is between '1' & '8'.

M·K_DIR [F4]

'M·K_DIR' allows you to create a fake/mock sub-directory name ("fake_name ›) which will then allow you to look at appropriately prefixed files (i.e., with the same "name") as if they were in a MAKE_DIR created sub-directory.

If you have selected the wrong device, then input MORE than ten 10) characters in the name to reset or simply press the (esc)ape key to exit.

The 'M·K_DIR' facility traps for duplicate filenames on the same medium.

GETTING A COPY OF QLATter

QLATter is really free if you send an e-mail message to me at:

alfeng@juno.com

I will send you a UUENCODED ZIP file which you must be capable of UUDECODing and UNZIPping at your end.
You will also receive a QLATter_txt file.
If you do not have e-mail, then please send $1.00 in the US or four (4) IRCs elsewhere to cover the cost of the disk and postage. Please specify disk size.
You can contact me at:
Al Feng
914 Rio Vista Circle SW
Albuquerque, NM 87105
■

Assembly Language Programming - Part 2

Norman Dunbar

The 68000 Instruction Set In part one, we learned some really boring stuff. Address modes are not what I would call interesting reading, and I suppose that most of you who are still reading this, would agree.
At this point, it gets worse. We are now going to delve into the instruction set of the processor.

Moving Data Around

The most common instruction in the entire world, is probably the MOVE instruction. It is actually wrongly named as it really does a COPY rather than a MOVE. The format of the MOVE instruction is:

```
MOVE source,destination or
MOVE.size source,destination
```

The data in source is copied to the destination. For example,

```
MOVE D0,D1
```

takes whatever data is in data register 0 (zero) and copies it into data register 1. How much data is moved? In this case. No size is specified so a word of data is moved from D0 to D1. As there is space for 2 words in each of these registers, which word is moved?

All instructions work from the 'lowest' end of the register towards the highest (with the exception of MOVEP - see below). So, in the above example, the lowest 16 bits of D0 are copied to the lowest 16 bits of D1. The data in D0 is not altered in any way whatsoever. The same cannot be said for D1 as the original data in D1 has been replaced - but only the lowest 16 bits. The highest word has not been altered.

If D0 contained $01020304 and D1 contained $11223344 then after the above move, D0 would be unchanged and D1 would contain $11220304. If the size of the instruction had been specified, as follows:

```
MOVE.B D0,D1
```

Then only the lowest byte of D1 would have been altered. In this case D1 would have contained $11223304 after the move. If the size specifier had been 'L' for LONG than the entire 4 bytes in D1 would have been overwritten by the 4 bytes from D0. After a long sized MOVE, both D0 and D1 would contain $01020304.

Because the move takes place into a data register the condition codes are affected. To copy data into an address register use the MOVEA instruction, but always remember that it does not affect the flags in the condition code register.

The changes that will take place every time a data register or memory location is used as the destination for a MOVE are:

- X flag is never affected. It remains as it is.
- N flag is set if the data moved was negative. If the data was positive, N is cleared.
- V is always cleared. You cannot move a value into a register that causes an overflow.
- C is always cleared for similar reasons.
- Z is set if the data moved was zero. It is cleared if it was any other value.

The MOVE instruction has many variations, most of them simple and easy to understand. These are:

MOVE as described above.

MOVE CCR - the size is always word although the upper 8 bits are ignored - effectively a byte sized move. The format of the instruction is:

```
MOVE source,CCR
```

Executing this instruction results in the condition codes being set as follows:

- X is set to bit 4 of source
- N is set to bit 3 of source
- Z is set to bit 2 of source
- V is set to bit 1 of source
- C is set to bit 0 of source

All the other bits are simply ignored.

MOVE SR - the size is always word and may not be specified in the instruction. This instruction copied the 16 bits of the condition code register to the destination. The instruction format is:

```
MOVE SR,destination
```

When the instruction has been carried out, the lower 16 bits of the destination contain a copy of the Status Register of the processor. The actual data in the status register is unaffected by the move.

There is a complimentary instruction to move data into the status register which is:

```
MOVE source,SR
```

Which takes the lower 16 bits of the source data and copies it into the status register. The lower 8 bits are used to change the flags in the CCR or Condition Codes Register (See MOVE CCR above). The SR is affected according to the lower 16 bits of the source data as follows:

T is set to bit 15 of source
S is set to bit 13 of source
III is set to bits 10, 9 and 8 of source
X is set to bit 4 of source
N is set to bit 3 of source
Z is set to bit 2 of source
V is set to bit 1 of source
C is set to bit 0 of source

The other bits are simply ignored. There is a slight problem, the instruction MOVE source,SR must be executed in Privileged mode or it will cause a 'Privilege Violation Exception' which on a normal QL will simply lock it up. (Exceptions are covered later on in the series.)

(Note: on the 68010 and up, the MOVE SR,destination becomes a privileged instruction. There is a new instruction MOVE CCR,destination which allows access to the CCR part of the SR. Programs written for the 68000 and 68008 may require to be re-written with this in mind.)

MOVE USP - A long sized instruction which copies data into the User Stack Pointer (USP) also known as A7. This instruction is also privileged and requires that the system is running in supervisor mode. The format of the instruction is:

```
MOVE source,USP
MOVE USP,destination
```

Both source and destination must be an address register. None of the condition codes are affected by this instruction.

Why does this have to be run in supervisor mode? Well, if not, a privilege violation exception will be generated and these instructions allow the operating system to set the value of a job's stack pointer.

If you remember, there are two A7 registers, one used for supervisor mode and the other for user mode. Only one can be in use at any one time. This instruction allows the supervisor to set the USP without affecting its own version of the A7 register. Not used much, if at all on the QL.

MOVEA - the contents (remember that word!) of the source is moved into an address register. This instruction is either word or long sized and does not affect the condition codes. The format is:

```
MOVEA.size source,An
```

Beware because if you move a word sized source, it will be sign extended to long (bit 15 will be copied into bits 16 to 31) before the data is copied into the address register.

For example:

```
MOVEA.W #$0001,A0
```

This will set A0 to $00000001 after the move. Bit 15 of the data is a zero so this is copied into all the upper 16 bits of A0. The lower 16 bits are simply a direct copy of the data.

```
MOVEA.W #$8000,A0
```

This will set A0 to $FFFF8000 after the move. Bit 15 is a one and this is copied into all the upper 16 bits of A0. The lower 16 are again a copy of the data.

Don't forget about sign extension!

MOVEM - a word or long sized instruction which allows you to copy data to or from a number of registers in a single instruction. The format of the instruction is:

```
MOVEM register_list,destination
MOVEM source,register_list
```

None of the condition codes are affected by this instruction.

The instruction is most often used to store a number of registers on the stack on entry to a subroutine, and to reinstate the original values on exit from the subroutine. The instruction stores the registers starting with D0, then D1 and so on up to D7, then the address registers are stored in order from A0 to A7 - assuming all registers are specified.

A register list takes the format of a starting register name, a hyphen then a finish register name. Another form is a start register name a slash and another register name. The two formats can be mixed to give almost endless possibilities. The following are all register list examples:

```
D1-D4
A0-A3
D1/D4-D7
D0-D2/D4/D7/A0-A3/A6
```

The hyphen means that all registers from the starting one to the finish one (inclusive) will be moved to the destination. The slash signals that there is a 'gap' in the register list. The above examples mean:

D1 and D2 and D3 and D4
A0 and A1 and A2 and A3
D1 and D4 and D5 and D6 and D7
D0 and D1 and D2 and D4 and A0 and A1 and A2 and A3 and A6.

The list can be specified in any order (unless the assembler rules differently) as each register

detected is used to set a single bit in a 16 bit word. This word is used by the processor to determine which of the registers are to be copied.

This instruction will be most often used in its Post decrement and pre-increment forms:

```
MOVEM.L D0-D3,-(A7)
MOVEM.L (A7)+,D0-D3
```

MOVEP - Probably the strangest instruction in the 68000 set. This instruction transfers data from a data register to alternating bytes in memory. The data is transferred from the data register starting from the highest 8 bits, then the next 8 bits and so on. This is a word or long sized instruction. The condition code flags are not affected. (I have never used or seen this instruction used on the QL.) The formats are:

```
MOVEP.size Dn,displacement(An)
MOVEP.size displacement(An),Dn
```

The size is long or word, Dn is any data register, An is any address register and the displacement is added to the address register to get the first address to be filled with data. An example might make things clearer. If we assume that D0 holds $11223344 and A1 holds the address $000200000 then the instruction:

```
MOVEP.L D0,0(A1)
```

copies the highest byte of D0 ($11) into address $20000, the next highest ($22) into address $20002, the next byte ($33) into address $20004 and finally the lowest byte of D0 ($44) into address $20006. Addresses $20001, $20003 and $20005 are not affected.

Had the displacement and A1 combined created an odd address then the odd addresses would have been filled with data and the even ones would not have been affected.

MOVEQ - This is a very useful instruction and you will see it used on many occasions in QL assembly language programs. It is the 'Move Quick' instruction and is used to quickly move any value between -128 and 127 into any data register. The value is sign extended to 32 bits or long sized and so fills the entire data register. The format is:

```
MOVEQ #data,Dn
```

The flags are affected by this instruction as follows:

X flag is never affected. It remains as it is.

N flag is set if the data moved was negative. If the data was positive, N is cleared.

V is always cleared. You cannot move a value into a register that causes an overflow.

C is always cleared for similar reasons.

Z is set if the data moved was zero. It is cleared if it was any other value.

Remember, only 8 bit values are allowed and these must be between -128 and 127.

A number of 68000 instructions have this 'quick' mode, but why is it quick? Let us compare the MOVEQ #0,D0 with its equivalent MOVE.L #0,D0. We simply see two different forms of what is effectively the same instruction, the QL's processor sees things a bit differently, as follows:

First MOVEQ #0,D0 is a 16 bit instruction in memory. MOVE.L #0,D0 is also a 16 bit instruction but it is followed in memory by a long word (32 bit) holding the data, in this case zero. This makes the MOVEQ instruction 3 times smaller than the MOVE.L one. As the processor has less data to fetch from memory, it takes less time to read the instruction and its data, therefore it is quicker. Looking at the 68008 timing chart, it takes the MOVEQ instruction 8 clock cycles to execute and the MOVE.L 24 clock cycles.

And that is about it for the 68008's MOVE instructions. This is probably the instruction with the most variants and as I said before, probably the most used instruction in any program.

Exercise

1. Write down the correct instruction which will copy 4 bytes of data from address $20000 into data register D7.
2. What is the fastest way to get the 8 bit value of 17 into all 32 bits of register D2?
3. What instruction would you use to copy the lowest 16 bits of register D1 into the lowest 16 bits of register D3? What happens to the data in D1 after the move and what happens to the data that is currently held in D3?
4. How would you place the lowest byte of D1 into a memory location which is 10 bytes further on from the address currently held in A0?
5. Why is the MOVE instruction 'wrongly' named?
6. What does a privileged instruction require before it can be executed?
7. What happens if a privileged instruction is executed in user mode?
8. How many data registers does the 68008 have and how many address registers?
9. What values are set in each of the condition codes wher the instruction MOVEQ #0,D1 is executed?
10. What values are set if the instruction executed was MOVEA.L #0,A0?

Answers can be found on the next page - please do not cheat!

Answers

1. MOVE.L $20000,D7
2. MOVEQ #17,D2 or MOVEQ #$11,D2
3. MOVE.W D1,D3. Nothing happens to the data in D1. The highest word on D3 is not affected but the lower word is overwritten by the lowest word from D1.
4. MOVE.B D1,10(A0) or MOVE.B D1,$0A(A0).
5. The MOVE instruction actually copies data from source to destination. It does not move it in the traditional sense of 'it was over there but it has been moved to over there'.
6. The processor must be in supervisor mode.
7. A privilege exception will be generated (and the QL will probably hang).
8. There are 8 data registers and 9 address registers but only one of the A7 'twins' can be used at a time.
9. The Z flag is set to one and all the rest are reset to zero except the X flag which is unaffected and keeps its previous value.
10. No flags are changed. They all keep their previous values.

Comparing Things

While all this talk of moving data around, be it in memory or within the processor's internal registers, is 'interesting', being able to move data is not much use if you cannot do anything with it when you have moved it. As the condition codes are affected by data movements we can sometimes determine the value of the data we moved. This is of course true only if we want to know if the value we moved was zero, or not zero, positive or negative but that's about as accurate as we can get using the MOVE instruction.

If we need to compare two values we will need to use the CMP family of instructions. CMP stands for 'Compare' and allows data to be compared against specific values, registers or memory contents.

The general format of the CMP instruction is:

```
CMP.size source,destination
```

The CMP instruction has the effect of carrying out a subtraction of source from destination without changing the destination at all. What it does change is the condition codes, and these will be set as follows:

X flag is never affected. It remains as it is.

N flag is set if the result was negative. If the result was positive, N is cleared.

V is set if the result caused an overflow otherwise cleared.

C is set if a 'borrow' was generated and cleared otherwise.

Z is set if the result was zero. It is cleared if it was any other value.

This instruction can be carried out in all three sizes - byte, word or long.

One of the common uses of this instruction, and perhaps the easiest to understand, is testing to see whether two values are the same. If they are then the result of the 'subtraction' of source from destination will always be zero. If the result is zero then the Z flag can be tested (somehow - we shall see later) and then some actions taken if it is set while others can be taken if it is not set.

The instruction:

```
CMP.L D1,D2
```

Will set the Z flag if the same value is present in both D1 and D2. If they are different, then the Z flag will not be set.

There are only four variations of the CMP instruction - unlike MOVE which has a few more. The first is simply CMP itself. This is used when comparing with a data register as in the above example. The source, however, can be any of the 68000 addressing modes - although you cannot compare an address register and a data register using the BYTE size. This means that:

```
CMP.W A0,D2
```

is a legal instruction, but that:

```
CMP.B A0,D2
```

is not. It is of course allowed that the data be POINTED to by an address register, as in:

```
CMP.B 0(A0),D2
```

Which compares the byte of data at the address held in A0 with the byte of data held in the lowest byte of register D2.

CMPA - is the form of the instruction used when comparing against a destination which is an address register. It is very similar to the CMP variation, but only word and long sized comparisons can be made. If the word size is used, then watch out for the old favourite pitfall of sign extension. Whatever word sized data is used for the source of this comparison will be sign extended up to a long word and then compared with the entire 32 bits of the address register.

This means that:

```
CMPA.W #$FFFF,A3
```

Would set the Z flag if and only if A3 contained the value of $FFFFFFFF but would not set it if A3 contained the value $0000FFFF. Beware. If at all possible, make your code explicit. SO if you want to test A3 as having $FFFF in its lower word, use CMPA.L #$FFFF,A3 instead of the word sized version.

CMPI - is the third variation and this one is used when testing any address mode destination (except PC relative or an address register's contents) against souce data which is, quite simply, a number. This variation can be used in all 3 sizes. The

format of the instruction is:

```
CMP.size #data,destination
```

If the destination is a data register, then the instruction is equivalent to the CMP instruction.

CMPM - is the final variation. It is used to compare one memory location with another. It can be used in all 3 sizes but can only be used in a single address mode - address register with post-increment. The format is always:

```
CMPM.size (An)+,(An)+
```

The two address registers are pointers to the memory addresses to be compared and after this instruction, the flags have been set according to the result of the 'subtraction' while both address registers have been incremented by 1, 2 or 4 depending upon the size of the data being compared.

Signed and Unsigned Numbers

Before we take a closer look at the condition codes and how we can use them to alter the flow of a program - that is, how we can implement loops, if then else etc, we need to take a braek and discuss the differences between signed and unsigned numbers.

When we MOVE some data into a data register the same number can actually mean two different things. Confused? You will be!

If we use an 8 bit number as an example, the data $FF can either mean 255 or minus one. In a 16 bit example, $FFFF can mean 65535 or -1 and in a 32 bit long word, $FFFFFFFF means 2^31 or -1. The important thing to remember is that it is you, the programmer, who decides which version is in use at any particular time.

Ok, how does it work? The 68000 family of processors can use signed or unsigned numbers. If the signed version is in use then the number will be either negative (less than zero) or positive (zero or greater). If unsigned numbers are being used then the value will always be positive. How can the processor tell the difference?

The answer to the question 'is this number signed or unsigned?' is either 'yes' or 'no' equivalent to one or zero in binary terms. This implies that a single bit can be used to hold the sign of the number and this is exactly how it happens. By convention the most significant bit of the number holds the sign. A one indicates that the number is negative while a zero indicated that it is not.

Those of you who are thinking ahead of me now might well be saying 'but surely using a single bit of the register will reduce the amount of numbers that can be represented by a factor of two?'. Not quite.

In binary, the numbers representing the hexadecimal values $00 to $0F will all fit into a half byte or nibble. A nibble is 4 bits and each bit represent a single power of two in the number.

Just as 1231 means (1 * 10^3) + (2 * 10^2) + (3 * 10^1) + (1 * 10^0), which is, (1 * 10 * 10 * 10) + (2 * 10 * 10) + (3 * 10) + (1 * 1) which is, 1000 + 200 + 30 + 1 which is the number we have at the start of all this, the same is true in binary.

The binary nibble 1010 is (1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (0 * 2^0), which is (1 * 2 * 2 * 2) + (0 * 2 * 2) + (1 * 2) + (0 * 1), which is 8 + 0 + 2 + 0, which is 10 in decimal with converts to $0A in hexadecimal.

All the possible values that can be held in an unsigned nibble are 0000 (zero) up to 1111 (15 or $0F) and conversion is a matter of adding up each power of two in the number. From the right we have 2^0 which is simply one. Then 2^1 or two and so on.

In an unsigned nibble the most significant bit (2^3) is used to hold the sign, so all numbers below unsigned 7 are positive while those 'above' 7 are actually negative and so are actually below 7.

If the highest bit was not the sign bit it would represent 2^3 or 8. To convert into a signed value simply negate the 8 to get minus 8, and add all the other bit values to it. Taking the same binary example of 1010 as above, this is now:

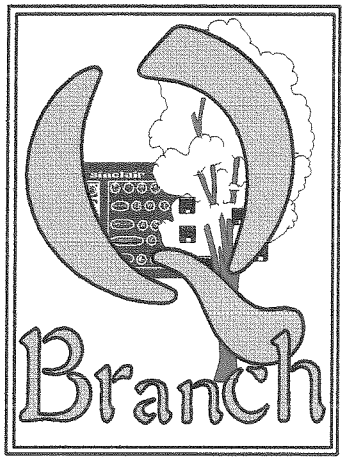
(-1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (0 * 2^0). This eventually gives minus 8 plus 2 which is minus 6. This now implies that for a signed number the range is -8 to +7 which is still a possible 16 values as with the unsigned version, just shifted slightly down the number scale.

That is the only difference between signed and unsigned numbers. The ranges of values in a byte are minus 128 to plus 127, in a word it is minus 32768 to plus 32767 and for a long word it is minus 2147483648 to plus 2147483647.

When dealing with signed numbers any number which has a 8, 9, A, B, C, D, E or F in the most significant digit (hex that is) is negative. All the rest are positive. I find the quickest way to find the equivalent negative value is to subtract from 2^number of bits. For example -1 in a byte is 2^8 -1 which is 256 - 1 which is 255. 255 in hex is $FF which is the 8 bit representation of -1. Similarly, -10 is 256 - 10 = 146 which is $F6. Use 65536 for 16 bit words and 4294967296 for 32 bit long words.

Enough for now. Just remember when coding a program in assembler that numbers can be two different values at the same time. You determine which one is appropriate at any one time. It is far easier to consider unsigned numbers all the time but this might not be applicable. Writing a

Q Branch



Q Branch at the Bank Volt

Shop hours ;
Mon - Sat
10am - 6pm
Wed
10am - 1pm

This is the new shop where you can buy all Q branch products as well as other computer hardware guaranteed to work with Aurora and superHermes. We have a functioning MinisQl and many other things. You can write to us there at this address :

Q Branch
The Bank Volt
6 Coronation Buildings
Ham Road
Worthing
Sussex
BN11 2NN

Or Telephone us in shop hours
+44 (0) 1903 200212
The other Q Branch numbers are still active.

SMSQ/E

Gold Card / Atari / QXL Version
£ 76.00

Various Atari versions : call for details

QPC v 1.43
£ 90.00 (£ 70.00 SMSQ/E owners)

We can now supply colour laptops with QPC already installed ! The systems are secondhand and vary considerably so you will need to call for details but a typical starting price for a 486 DX 70 Toshiba with 350 Meg hard disk and 12 Ram complete with QPC and Carrying case is

£ 500.00 (+postage)

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3
£ 15.00 ea / ANY 2 PROGRAMS £ 25.00 / ALL 3 PROGRAMS £ 35.00

!!! NEW VERSIONS !!!

THESAURUS V4.01 / SOLVIT PLUS 3 v 3.00

Pointer drivens versions of Geoff Wicks popular programs.
(needs Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)

Still the same price £ 15.00

Upgrades from previous versions £ 2.50 + S.A.E.

PROGRAMMING

QD v 9.15	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1 & 2 (together)	£ 33.00
Easyptr pt 3 (C library)	£ 16.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

NEW PRICES !!!

Text 87

£ 79.00

Typset 94 £ 29.00
Fountext 94 £ 39.00
2488 drivers £ 29.00
Epson ESC/P2 drivers £ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handliing the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written. WATCH THIS SPACE !

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread v1.44	£ 66.00
Cueshell 2	£ 30.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00
LDUMP	£ 26.00
Q Count	£ 25.00

Youv'e been asking for it and here it is !

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !
£ 40.00
+ postage

Q Branch

Feeli~ g out on a limb ?
Reach out for Q Branch
Suppliers of Quality QDOS/SMSQ products
Hardware and Software

Package deals

QXL II + SMSQ/E	£ 220.00
Aurora + SMSQ/E	£ 160.00
Aurora + SMSQ/E + Super Gold Card	£ 330.00
Above package + Monitor	£ 380.00

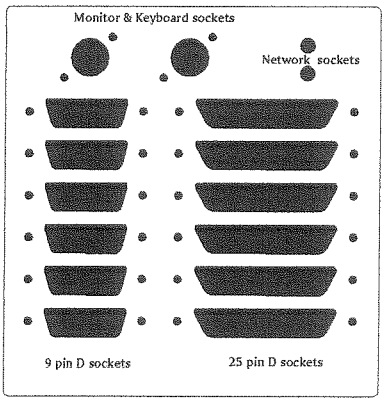
Hardware

QXL II	£ 180.00
Super Gold Card	£ 160.00
Recycled Gold Card	£ 60.00 *
Aurora	£ 100.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
The 'Braquet'	£ 16.00

* when available.

NEW PRICES !!!

The MC - Plate



ONLY £ 6.50 !

All the socket holes you will ever need on the back of your tower case. Just rip off the silly PC thing and screw on this one. No cutting needed. What could be easier ?

ProWesS

ProWesS	£ 48.00
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Fontpack	£ 60.00
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

Q Branch

The Big Q News this is issue is that we are now operating from a shop in Worthing called the Bank Volt. 5 minutes walk from the seafront and 10 minutes walk from East Worthing station so we are easy to reach - even if you come by boat ! The advantage of the shop is that we will stock some standard PC items but, if you want disk drives / keyboards / etc for your QL systems we will be able to advise you. See next page for details.

We have almost sold out of QXL IIs and Super Gold Cards now - only one or two left at the time of writing so hurry if you want one.

The SBASIC / SuperBasic reference manual is out now. See next page for more details. Shipping costs for this item are high so the best plan is to order it and pick it up from the nearest QL Workshop.

We also have a few SVGA Aurora monitors. See below for details.

Tel +44 (0) 1273-386030 fax +44 (0) 1273-381577
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch Programs

The Knight Safe 1 - standard version	£ 30.00
The Knight Safe 2 - with compression	£ 35.00
Q - Count Pointer driven home accounting	£ 25.00
Q - Route v1.07 Route finding programme	£ 25.00
Page Designer 3 S Pointer driven desktop publishing for big screens	£ 40.00

2nd user 14" SVGA Colour Monitors ! (suitable for Aurora)

Ready for use and giving all screen resolutions from 512 x 256 to 1024 x 576

£ 40.00 + shipping

VISA

We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.

MasterCard

program to record the number of sheep jumping over a fence need never use signed numbers, while the amout of money in your bank account probably will. Just remember to be consistant.

Testing Condition Codes and Branching

As you may remember when data is MOVED into a register or memory address, certain condition codes are set or unset. These codes can be used, along with the results of a CMP instruction and/or the discussion of signed and unsigned numbers above, to determine program flow. To change the flow, we use the branch instruction also known as Bcc or Branch on condition code. The general format of a Bcc instruction is:

```
Bcc label
```

The label part defines where the branch will be to (the destination) and is an offset from the current program counter and of course may be positive or negative.

A branch instruction is equivalent to a SuperBasic GOTO command. Much frowned upon by purists, but useful in certain situations. Never say 'Never use a GOTO' because in assembly language you almost always have one!

There are a number of 'branch' instructions that look at the condition codes and change the course of your program according to what they find. There are 14 of these ands some appear remarkably similar to others. They are:

BCC - Branch Carry Clear - unsigned. The branch is executed if the carry flag is not set - ie zero.

BCS - Branch Carry Set - unsigned. The branch is executed if the carry flag is set - ie one.

BEQ - Branch Equal - signed and unsigned. Branch only if the result of the last operation caused the zero flag to be set. MOVEQ #0,D0 for example.

BGE - Branch Greater or Equal - signed. Branch if the last operation resulted in a signed number that was zero or greater.

BGT - Branch Greater Than - signed. Branch if the last result was greater that zero.

BHI - Branch Higher - unsigned. Branch if the last result was greater than zero.

BLE - Branch Less or Equal - signed. Branch if the last result was zero or less.

BLS - Branch Low or Same - Same as for BLE but unsigned.

BLT - Bacon Lettuce & Tomato (only kidding!) Branch Less Than - signed. Branch only if the last result was less than zero.

BMI - Branch Minus - signed. Branch if the result of the last operation was negative. Ie less than zero but not including zero.

BNE - Branch Not Equal - signed and unsigned. Branch if the last operation resulted in a non-zero outcome. CMPI.L #1,D1 if D1.L is not holding the value 1.

BPL - Branch Plus - signed. Branch if the result of the last operation is positive ie zero or greater.

BVC - Branch oVerflow Clear - branch if the last operation left the V flag unset.

BVS - Branch oVerflow Set - branch if the last operation left the V flag set.

There is one more branch instruction that does not care about the flags, this is the BRA or Branch unconditionally instruction. It is the most like a GOTO instruction as that is its exact purpose - goto some other place in the program.

If the displacement value will fit into a single byte (-128 to +127) then a 'short' branch will take place. This entire instruction fits into a single word. If the displacement is zero, then this would normally indicate a short branch to the next instruction in the program. As this is where the PC is pointing anyway the zero displacement is used to signify a long branch and the word following is used as a 16 bit displacement allowing relative values between -32768 to +32767.

The short branch is written as Bcc.S with the dot and 's' indicating the shortness. Most assemblers default to the long branch which adds 2 bytes to your program for every Bcc instruction in it. I find the 'best' way to reduce the 'wasted' bytes is to make all branches short and the assembler will reject those which are out of range.

One of the most confusing aspects of assembly language programming for new and experienced coders alike is 'which are the signed and unsigned tests?' I always have to look it up and I have never found a place where all the tests are listed together with the signed and unsigned comparisons. You won't have this problem as I have listed them all below.

Test	Signed	Unsigned
Greater Equal	BGE	BCC
Greater Than	BGT	BHI
Equal	BEQ	BEQ
Not Equal	BNE	BNE
Less Equal	BLE	BLS
Less Than	BLT	BCS
Negative	BMI	Not applicable
Positive	BPL	Not applicable

In the above description of the Bcc instructions I state, for example, that the BNE instruction will branch if the last result was not zero. This is not

quite the case. If I had just loaded a data register with some value which was not zero then the branch would be taken, as in the following fragment of code:

```
MOVE.L (A0),D1
BNE.S Somewhere
```

If, on the other hand, I was comparing two registers then the branch would have been taken if they did not have exactly the same contents:

```
CMP.L D3,D4
BNE.S not_equal
BHI.S greater
```

So you can see that there are more ways to use these conditional branches. Bear in mind, however, that the CMP is simply a subtraction with the result 'thrown away' and it is that result that is being checked. One other area of confusion is which register is greater in the BHI instruction above?

In a CMP instruction it should be read as Destination CMP source. If this is followed by a Bcc then it means branch if the destination is ‹condition› source. So in the above code fragment, we will branch to the label 'greater' if and only if D4 is greater than D3.

There are other instructions that affect the flow of a program and these are the 'looping' constructs or DBcc as ther are written. These are the 'Decrement and branch UNLESS condition. Confused? All will be revealed next time.

■

Build an Atari ST SMSQ/E computer

Donald Walterman

I am always interested in different hardware used with QL computers. A while back, I decided to see what SMSQ/E looked like running on Atari computers. It's now one 1040ST and three MegaST's later.

While I still know very little about GEM and the Atari, I have had great success using these machines with SMSQ/E. I'm considering writing a few articles about how to find, upgrade and use these machines with SMSQ/E.

Here are a few general comments to give you and idea of what is available in the United States. I don't know how widely available Atari hardware is in other countries.

I've bought Atari's for as little as $25. The Atari platform sold in big numbers in the United States so there are many available. All Atari ST models have at least a full 68000 8 MHz processor.

The Atari ST has a good keyboard, mouse, built in floppy, built in DMA bus for hard drives, midi, serial ports and a parallel port. The DMA bus can easily be upgraded to a full SCSI bus allowing you to use standard SCSI peripherals. The best Atari to find is the TT. This is a great system with a 68030 processor at 16 or 32 MHz. Avoid the Atari Falcon. This was a great Atari but SMSQ/E will not run on it.

Basic Comparison Chart

Classic QL hardware family	Atari ST/STE family
68008 on QL · 7.5 MHz	520, 1040 and Mega use 68000 · 8 Mhz
68000 on Gold Card	TT uses 68030 · 32 Mhz
68020 on Super Gold Card	many processor upgrades were made
no classic processor hardware currently in production (GoldFire is coming)	Atari is no longer in business A limited number of processor/memory upgrade boards are still available
floppy drives	floppy drives
720k drives on QL max 4	720k drives on ST max 2 (very old 360k)
DD, HD, ED on GC, SGC	DD, HD on upgraded ST/STE/TT models
hard drives available	hard drives available
Qubide for IDE drives	DMA drives from Atari now obsolete
JFC for MFM - obsolete	current SCSI drives can be used with
Miracle drive obsolete	ICD Link/2 which is still available

monitors
RGB, composite or TV
Aurora supports SVGA
standard display 512x256
Aurora allows 1024x512

monitors
requires Atari SM124 monochrome
VGA in monochrome with custom cable
standard display 640x400
standard QL monitor can be used with
ATARI QL emulator - various higher resolutions in MODE 4

The main disadvantages with the Atari are: The floppy drives with a max of two drives. Most systems only support 720k drives unless you do a hardware upgrade. The upgrade will let the Atari ST support HD drives as well.
The second item is the monochrome monitor requirement. Atari STs have a number of modes but you must use the high resolution mode for SMSQ/E. This is only available in monochrome. This resolution is better than the original QL though. The Atari SM124 is a nice crisp monitor that reminds you of the original MAC. An alternative is to use a standard VGA monitor with a special cable that ties all colors together. You get the same resolution still in monochrome but you aren't tied to the SM124 monitor.

[Editor's comment: in Europe, the SM124 requires 71Hz refresh rate - most VGA monitors handle 60Hz only unless they are MultiSync or Multiscan - so beware!]
The main advantages are:
Solid one piece box with quality construction, easy SCSI device support, parts and upgrades still available from dealers. Low entry cost for basic system. Even low end 520ST has 68000 8 MHz processor with DMA bus.
The Atari 520ST doesn't leave you much memory to work with under SMSQ/E. A realistic minimum Atari is the Atari 1040ST. I would recommend looking a little harder for a Mega ST2 or Mega ST4. If you can find a TT, buy it! I'm still looking for mine...
Well, that gives you an idea of what I'm thinking about. Please let me know if there is any interest in this topic. It will be heavily biased toward hardware. I would like to cover the hardware upgrades I've done and what problems I've run into. I also intend to cover catalogs, dealers and internet resources I've found. There also is a small list of essential software needed to get the most from the hardware.
I'm sure there are many people in Germany that could supply much more information. I'd welcome any corrections or further discussion. Maybe this will encourage someone to write a more detailed article on the Atari ST and SMSQ/E.
Please feel free to send comments to QL Today or directly to me at
dwalterm@ix.netcom.com
■

QDOS on Q40

Mark Swift

For the past few weeks I have been busily porting Amiga-QDOS onto the Q40 computer. The Q40 is a hardware replacement for the QL with a FAST 68040 processor, QL screen modes (plus others at hi-colour/hi-res), IDE, floppy, serial, parallel, sound, etc... Here is a diary of events, hastily put together, but more-or-less as things happened during the project's continuing development.

The beginning

In April of this year Simon Goodwin phoned me and asked if I was interested in porting Amiga-QDOS to the Q40. He said that Peter Graf (the man behind the Q40) was looking for developers and that there was a possibility of a loan of a prototype Q40 board. I replied that I would be willing to look into porting Amiga-QDOS, but couldn't promise any results. I added that I certainly wouldn't say no to a loan of a prototype Q40. Simon passed on my comments to Peter.

The offer I could not refuse

Later, Peter Graf e-mailed me with some hardware details, and at the beginning of May stated that he had a prototype for me 'if I was interested.' I replied 'DEFINITELY yes, I'd be VERY interested'. The offer was that he would send me a 40 MHz Q40 Mainboard with 16 MB RAM for free, 'if you port (Amiga) QDOS including Keyboard, Floppy, Serial and Clock until 01-11-98'. Two weeks later I e-mailed Peter - concerned that I still hadn't received the board.

The missing parcel

Thus began a very frustrating couple of weeks... Peter replied to my e-mail stating that not receiving the board 'was serious' since he had sent it a week before by airmail for a quick delivery. He said that he hoped that I would receive it soon since he was he was to leave for vacation, and it would be two weeks before he could investigate the lost parcel. I checked at the Post Office, no knowledge of any parcel. Time passed... still no board. I re-checked at the Post Office, they still claimed no record what-so-ever. They said that if the postman had called while I was out he would have left a calling card - no card - no parcel.

On 23 June Peter wrote that the parcel had finally been returned to him, 5 weeks after he had sent it. It had been returned by the British post - the reason for the returned parcel being: "Gone away / House empty" - my faith in the Post Office was severely dented. Peter said that the prototype was to begin its next journey, and should arrive within a few days 'if we are lucky'. He stated that I could now keep the board for evaluation until 01.01.99 instead of the 01.11.98 and that his offers for a free Q40 still stood. On Fri. 26 June, with much relief, I finally received the board.

What is it?

The board is very small. There is the 68040 processor, two SIMM slots that can take up to 32Mb of memory and two EPROMs that contain the utility software. There are also two expansion slots, one of which is occupied by a joint floppy disk/IDE harddisk card. The following ports are available on the main Q40 board:

LINE OUT,
SPEAKER OUT,
VIDEO OUT,
KEYBOARD

The expansion board is a standard ISA IDE/floppy controller for a PC and has the following ports:

IDE HDD,
FLOPPY,
2 SERIAL,
1 PARALLEL,
1 JOYSTICK PORT.

The Q40 board also has an in-built clock and 2040 bytes of non-volatile RAM. Since I have not gone the PC-route I had to borrow a PC keyboard. I already had a spare PC power supply that my brother used to use before he put his Amiga in a tower, and I was able to make use of a standard 15 inch multiscan monitor (which I again borrowed). I was hoping to use my Apple Mac 14 inch monitor - but this is not multisync and was too puny to cope with the Q40 display modes. At this time the board is not housed in a case. If I am lucky enough to keep the board I will have to buy the following:

MINI-TOWER(£25),
KEYBOARD(£10),
MULTISYNC MONITOR(£120).

What does it do?

The board I received contained a utility ROM that allows you to test the various bits of hardware. It also allows you to upload (via the serial port) some demo pictures, or your own pseudo ROM. The first thing I did was to build a serial cable to connect to my QL - so that I could view the demo pictures. These are 512x256 and 1024x256 at 64K colours and are very impressive. Eventually, having grown tired of having a super-QL with no operating system, I set about porting QDOS.

QDOS-Classic

The initial idea behind this QDOS port was simply to provide a means of running OLD software on the Q40 in a more-or-less compatible way. I was hopeful that someone else would be supplying an o/s (operating system) more appropriate to the Q40's modern-hardware-status. However, having an o/s where the sources are freely available can be very useful. This gives everybody the opportunity to improve and extend it. Since it was no-longer a specific Amiga implementation of QDOS, I had to rename the project. I have since named it QDOS CLASSIC - in this case QDOS CLASSIC v3.25 (beta) for the Q40. When I update the Amiga sources, that implementation will be called QDOS CLASSIC v3.25 for the Amiga.

QDOS CLASSIC is NOT an emulation. It is written in 680X0 machine code to run directly on Motorola 680X0 processors - so it is FAST. Much faster than a current pentium system pretending to be a 68000 can hope to be. QDOS CLASSIC is a like a JS ROM, but with all the hardware bits removed. The idea is to have a QDOS ROM that runs on any 680X0 system, and to have all the hardware specific parts implemented as external add-on ROMs. Thus there are ROMs for the CLOCK, the KEYBOARD, etc... With nothing hardware-specific in the main ROM to crash-out the system, development time was cut considerably. I was able to implement specific hardware support in logical steps that could be tested and implemented on an individual basis:

STEP 1 -
try it and see if it works

The first thing I tried was to upload the current Amiga-

QDOS ROM just to see if it did anything, it didn't - it didn't even get to the F1/F2 screen. No surprise there.

STEP 2 -
persistence
Having re-examined the code and made a few adjustments I tried again with more confidence. Still nothing. Perhaps this was to be expected, in the initial stages of a project there can be many unexpected bugs. For this reason it pays to streamline the testing process. To make life easier, I built myself a split serial cable. Input to the Q40 comes from my QL and output from the Q40 goes to my Apple Mac. At the moment I am developing and assembling the sources on my Amiga, transferring the code onto QL floppies, uploading this from my QL to the Q40 (at 9600 baud) and sending debug messages to my Mac (at 115200 baud). In fact, in the initial stages of the project (when nothing worked) I single stepped the whole of the ROM and captured the output to a file on my Mac. This worked out at around 9Mb and took all night to transfer, but since it was automatic I was able to get some sleep. After many hours examining code, I finally got the F1/F2 screen to show. Clearly, to get this far a lot of the code had worked.

STEP 3 -
frame/polled interrupts
At this stage there were no keyboard routines, so it wasn't possible to press the F1 or F2 keys - instead I commented out the F1/F2 code just to see how much further it got. It went all the way to the split screen, but with no flashing cursor. I then realised that I had to write the routine to recognise frame (polled) interrupts. When I had done this I finally got a cursor that blinked healthily - which was a good sign since it proved that the system was actually running.

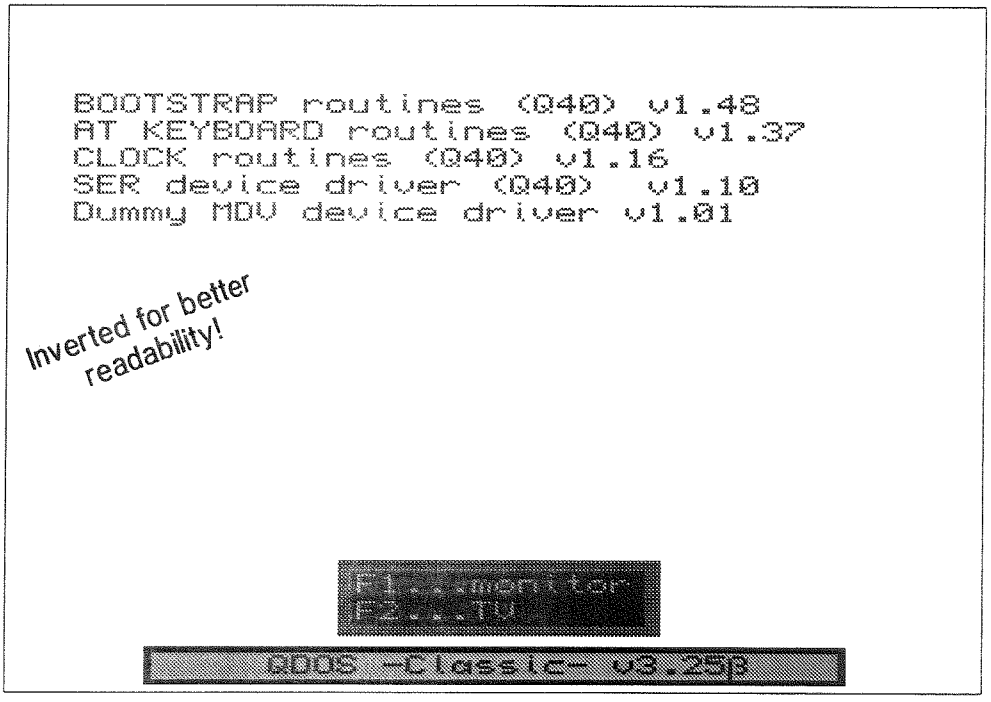
STEP 4 -
keyboard
The Q40 uses a standard PC AT-keyboard which connects to the Q40 via a 5 pin DIN connector. At low-level the AT keyboard gives you a keycode when you press a key, and the same keycode preceded by a 'release' code (240) when you release the key. Also there are 'special' keys that return multiple keystrokes. For example, the BREAK key returns 8 keycodes:
225,20,119,225,240,20,240,119
I had managed to find quite a bit of information about PC keyboards while I was waiting for the prototype to arrive - so it didn't take too long before I had written the first draft of the keyboard routines. This meant that I now had a working, albeit minimal system and on Wednesday 22 July I typed in my first program:

```
10 PRINT "Hello World!"
20 GOTO 10
```

The BASIC interpreter worked! - though I soon found that there was a bug in my CTRL-SPACE routine so that I couldn't BREAK out of loop... However since it was 3:30am and since I had to go to work in the morning, I decided to call it a day and get some sleep. I fixed most of the major keyboard bugs the next evening.

STEP 5 -
clock
Now I set my mind to the in-built clock. In the end the clock routines turned out to be pretty straight-forward, especially since I was now able to PEEK and POKE the hardware directly from BASIC. After getting the in-built clock working, DATE$ told me that it was 1:00am and time for a rest.

STEP 6 -
name it and send out a beta copy
I needed to have something to print on the banner of the F1/F2 screen - so it was at this stage that I renamed the project from Amiga-QDOS to



QDOS 'CLASSIC'. QDOS CLASSIC seemed an appropriate name since it is based around the now classic JS ROM - a fairly stable and compatible ROM set. I have since realised, that I probably read the term 'classic' in Adrian Ives' article in the July/August edition of QL today when he described his black-box as his 'QL-Classic'. Anyway, on Friday 24 July I e-mailed Peter Graf the first working beta.

How fast is the Q40?

Having got a working BASIC interpreter, I could now test how fast the Q40 really was - which all depends on the processor cache settings. QDOS CLASSIC includes three BASIC keywords that determine how the caches are utilised.

SERIALIZED sets the memory to be non-cachable.
WRITETHROUGH sets the memory to be cachable. Also, during a write, memory is immediately updated from the caches.
COPYBACK sets the memory to be cachable, but memory is only updated from the caches when absolutely necessary (i.e when the caches are full). Enabling COPYBACK on 68040 & 68060 machines gives substantial speed increases.

As you can see, on these tests with copyback enabled, the Q40 is around 40 times faster than a standard QL. You should note that the version I was sent is a 40Mhz 68040. There is a slower 33Mhz version and a much quicker 68060 version running at 50Mhz. There are also plans for 68060 versions running at 66Mhz (full) and 75MHz (EC & LC only).

The benchmarks I used were QSBB_bas from the QLAY distribution.

CPU	PRINT	FUNCTION	STRING	CONFIGURATION
Q40 68040/40	16700	14840	20980	QDOSclassic (serialized)
Q40 68040/40	24760	24820	37420	QDOSclassic (writethrough)
Q40 68040/40	30420	34220	46860	QDOSclassic (copyback)

The same tests on an original QL give the following timings:

CPU	PRINT	FUNCTION	STRING	CONFIGURATION
QL original	980	840	1100	128k JS-ROM

What's next?

SER support - then PAR - then FLP and - then sound, probably in that order.
B.T.W. QDOS CLASSIC sources are freely available to anyone who's interested.
■

QDOS Bugs - Part 1

Mark Knight

The various QL ROMs contain several bugs. Mark tells you about most of the bugs he is aware of, and also shows you ways to avoid the individual problems.

As a programmer it's an irritating fact that eliminating bugs in my code is not enough: I often have to work around bugs in other peoples' too. Programs that work fine on my QL may fall over on another system simply because they run into a bug in the users' system. Part of the solution is testing and I have a friendly beta-test team who kindly test much of what I write on their systems and report, sometimes in painful detail, when it goes wrong.
A large part of the process of avoiding the bugs in other people's code is to keep yourself informed, in other words collect bug reports. This enables you to more readily work out why, for example, the carefully written BOOT program for your magnificent new program works on your MG ROM QL but won't run on somebody else's AH system. I have been collecting bug reports for a number of years and I know of only one person who has a more comprehensive list of the bugs in the official QL ROMs.
I have a much smaller and less complete list of the bugs in Minerva 1.97 ROM, I'm told there are others and that several of the JS ROM bugs should also be in the Minerva list too, though I don't know which. Both of these lists are below and I hope others find the list useful, I certainly have. The bug that has caused the most trouble over the years is number two on the list, the simple fact that AH ad JM ROM systems won't recognise extension keywords used by the same SuperBASIC program that loads them into the system.
This means that you can't do things like this on early ROM versions:

```
100 TK2_EXT
110 WDIR "flp1_"
```

...because the TK2_EXT command loads the WDIR command into the system, and the rule is you can't use a keyword in the same program that loads it into the system. You have to do this instead:

```
100 TK2_EXT
110 LRUN "flp1_NextBit_BAS"
```
...and the next bit is a separate program:
```
100 WDIR "flp1_"
```
If you are working from the command line on one of these systems you can type NEW instead, like this:
```
TK2_EXT
NEW
WDIR "flp1_"
```
This bug is the reason why many BOOT programs for QL applications are split into BOOT and BOOT2. BOOT loads the toolkit, and BOOT2 uses some of the keywords in it, perhaps to set up the screen or some environment variables for the application about to load. If you have a JS or later ROM you can usually renumber and MERGE BOOT and BOOT2 without trouble, though examine them carefully first and if in any doubt don't.

I have laid out the list in a simple format; first the bug is described, and I've given them all classifications to help you work out if they will affect your compiled or native machine code programs. A bug described as a BASIC bug should affect only programs running under the SuperBASIC interpreter or one that calls interpreter routines. Most compiled SuperBASIC programs don't show these bugs as they are effectively now machine code, but some do as the compiler may set up code to call the standard interpreter routine. Bugs marked as SYSTEM bugs will affect any running program, even one in machine code, unless evasive action is taken by the programmer.

A brief suggested "fix" or evasive action is also included for each bug, if others know of better ways of dealing with these problems perhaps they will be stimulated to write to QL Today and let the rest of the programming community know.

Certainly this habit of collecting bug reports has helped me recently to write a fractal program that (so far) runs on every system it has been tried on, QLs with JM, JS and MG ROMs, SMSQ on a QXL, SMSQ/E on QXL, various Atari emulators and QPC. This portability is partly due to programming with known bugs in mind and also due to collecting information about QL compatible systems generally. The article by Dilwyn Jones in the September/October 1997 QL Today helped as well, as it told me how to find out how big the screen is on any system - a handy bit of information that! I am also indebted to Rich Mellor for information on some of the bugs in Minerva 1.97.

This is not a full list of all the bugs I have descriptions for, but includes all those properly documented. Some I have such vague descriptions of that I hesitate to pass on what may be inaccurate reports, so perhaps others can write and let QL Today know. Some oddities or properties of the QL ROM versions that are often described as bugs are not on this list because they are not bugs; they are documented features.

An example of this is the fact that CHR$ takes numbers outside the range 0-255: Sinclair in fact documented this and Turbo intentionally reproduces it when compiling programs for compatibility. POKE, POKE_W and some other commands also work with numbers outside their "proper" range, using MOD to bring the numbers down to size or else simply converting the number to a 32-bit integer and ignoring the unwanted 16 or 24 bits. Perhaps a better example of a bug that isn't is that RESPR does not work on most systems once there are some jobs running - it isn't supposed to, so this isn't a bug.

If you know of a bug in any QL ROM, Sinclair or Minerva 1.97, please let QL Today know about it so others can either writte a corrective patch or work around it when programming. If all QL programmers help with collecting information on bugs both users and programmers will have fewer headaches.

Known Bugs in official Sinclair QL ROM versions

1. Slicing a slice of a string array (not a string) gives wrong result (BASIC). AH JM
   Fix: Assign to a temporay variable or compile with TURBO.
2. SuperBASIC does not recognise new keywords if they are part of a SuperBASIC program already loaded (BASIC). AH JM
   Fix: Use NEW or LRUN etc. in between loading and using new keywords.
3. INPUT cannot handle more than 128 characters (BASIC). AH JM
   Fix: Don't try to INPUT very long lines on these machines.
4. PI, BEEPING, VER$ and DATE do not reject parameters (BASIC). AH JM
   Fix: Minor bugs, hardly likely to bother anyone.
5. CALL used from a SuperBASIC program bigger than 32k may crash the system as it often calls the wrong address (BASIC). AH JM
   Fix: Compile with TURBO or load a patch routine. Systems with Toolkit II active not affected as it contains such a patch, and so does Turbo Toolkit.
6. String or integer variables may be input as SELect variables even though SELect cannot

QUANTA

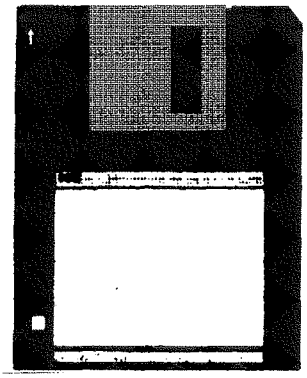
Independent QL Users Group

Worldwide Membership is by subscription only, and offers the following benefits:
Monthly Newsletter - up to 40 pages
Massive Software Library - All Free !
Free Helpline and Workshops
Regional Sub-Groups. One near you?
Advice on Software and Hardware problems
Subscription just £14 for UK members
Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our FIFTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407

work with them (BASIC). AH JM

Fix: Compile with TURBO or Q-Liberator using IMPLICIT% or IMPLICIT$ with TURBO, DEF_INTEGER with Q-Liberator (BASIC).

7. Use of a PROCedure parameter as a SELect variable inside the PROCedure will fail with "bad name" error (BASIC). JS

   Fix: Assign to another variable before using with SELect or compile with TURBO or Q-Liberator.

8. Use MODE command and some default windows (i.e. SCR or CON windows) may be redefined and the ink and paper colour both set to black (SYSTEM). AH JM

   Fix: Define windows explicitly with parameters slightly different from the defaults or avoid use of MODE once windows are open.

9. SEXEC or SBYTES do not work properly when "bad parameter" errors occur, leaving an empty file in existence (SYSTEM). AH JM

   Fix: Avoid passing odd parameters to SBYTES or SEXEC.

10. String compares may match a "." character equal to a "0" character as "." is treated as an embedded number (SYSTEM). AH

    Fix: Compare codes rather than characters if possible.

11. Floating point arrays are limited to a total of 65536 elements (BASIC). AH

    Fix: Compile programs with TURBO or Q-Liberator.

12. OPEN_IN and equivalent TRAP calls will open the file at the start of the header rather than start of file if it is already open to another task or channel (SYSTEM). AH

    Fix: Avoid letting programs share files on affected systems.

13. Calling a SuperBASIC PROCedure or FuNction which has been deleted can crash the system if the routine had a line number greater than any now existing (BASIC). AH JM JS MG

    Fix: During development you can keep a line "32767 STOP" at program end, or SAVE and reLOAD after deleting a PROCedure or FuNction.

14. CURSOR command with 5 parameters (e.g. CURSOR#Ch,a,b,c,d) fails or ignores last parameter (BASIC including compiled). AH JM JS

    Fix: Restrict this sort of use of CURSOR to #1 so the other parameters are picked up (#1 is default and need not be specified).

15. CLS or PAN to end of line clears too much if used in very narrow window (too narrow to take even one character). (SYSTEM). AH JM JS

    Fix: Don't use absurdly narrow windows.

16. Expression in a DATA statement starting with a bracket will cause the rest of the line to be ignored by the interpreter (BASIC). AH JM JS

    Fix: If you must use expressions in DATA, start them "0+(" if there is no other way of avoiding a leading bracket.

17. Comparing strings often gives wrong results if characters greater than CHR$(127) are involved (SYSTEM). AH JM JS MG

    Fix: Compare codes rather than characters if possible.

18. GO SUB in a single line FOR loop will terminate the FOR loop as if it were an END FOR (BASIC). AH JM JS MG

    Fix: Don't use this construction. Compiling with TURBO will fix the bug but you shouldn't be using GO SUB in SuperBASIC anyway.

19. Trying to access mdv8_ will corrupt memory and confuse the system even if mdv8_ exists (is this likely?) (SYSTEM). AH JM JS

    Fix: Don't use more than 7 microdrives! I would love to know who it was who discovered this bug...

20. Trying to EDIT after breaking into a program or after an error or STOP inside a PROCedure or FuNction may give nasty errors, including "not implemented" followed by presenting wrong line for editing (BASIC). AH JM JS MG

    Fix: Break out of the EDIT and try again or system crash will follow.

21. Break cannot escape from a one line recursive PROCedure (is anybody mad enough to use them?) (BASIC). AH JM JS MG

    Fix: Don't use them.

22. READ or INPUT to a substring of an as yet undimensioned string or array may halt SuperBASIC with no error message (BASIC). AH JM JS MG

    Fix: Don't let it happen.

23. Double declaration of a LOCal, or a LOCal which is also a parameter passed to the routine may crash the SuperBASIC interpreter (BASIC). AH JM JS MG

    Fix: Don't make multiple LOCal declarations. Compile with TURBO and the parser will report such declarations.

24. Serial transmission can lose characters or become seriously spasmodic if several CPU bound jobs are running (SYSTEM). AH JM JS MG

    Fix: Suspend or unload such jobs while using serial ports.

25. COS with arguments greater than 16384*PI gives nonsense results (SYSTEM). AH JM JS MG
    Fix: Don't use absurd arguments with COS.
26. WHEN as an immediate command can crash the system (BASIC). JS MG
    Fix: Compile programs that must use WHEN.
27. Deleting a WHEN statement can crash the system (BASIC). JS MG
    Fix: Compile programs that must use WHEN and edit them in an editor.
28. WHEN interacts badly with RENUM, which can crash the system. (BASIC). JS MG
    Fix: Compile programs that must use WHEN.
29. WHEN variable sometimes gives inappropriate double hits or "bad name" errors, this can crash the system (BASIC). JS MG
    Fix: Compile programs that must use WHEN.
30. NEW and CLEAR do not clear the WHEN ERRor flags (BASIC). JS MG
    Fix: Compile programs that must use WHEN and don't run them under the interpreter.
31. SVT.VMOD system variable is changed by a MODE call (SYSTEM). AH JM JS
    Fix: Read it before any MODE command has been used.
32. Passing a slice of a string array (not a string) to a PROCedure or FuNction which uses PRINT will cause several bytes of memory to be lost until the next CLEAR or NEW. (BASIC). AH JM JS
    Fix: Assign to a temporary variable before the call or compile with TURBO.
33. Attempts to access a directory device with less than 1k free can crash the system instead of giving an "out of memory" error (SYSTEM). AH JM JS
    Fix: Keep a close eye on free memory.
34. VER$ does not allocate stack correctly and can crash the system or give nonsense results if used as part of an expression (BASIC). JS
    Fix: Assign it to a string before using the result in an expression.
35. More than 9 LOCals or parameters in one PROCedure or FuNction can crash the system (BASIC). AH JM JS
    Fix: Compile the program with TURBO or Q-Liberator.
36. DATA can be renumbered as if it were a line number if a DATA statement is used on the same line as RESTORE (BASIC). AH JM JS
    Fix: This is a daft construction anyway; don't do it.

■

All the World's a QL Stage - Part 1

Doug LaVerne

Adventures with the Internet, the QL, QTPI, and an ISP (Internet Service Provider).

Introduction

In 1995 I wrote an article for the now defunct "International QL Report" (ISSN 1078-5787) about going "around the world in 80 ways" via a basic QL. I'm doing it again, now that at Bedford QL NA '98 Don Walterman and John Impellizeri fixed cats' damage to QL system components.

It is somewhat surprising to see claims in print still that it is difficult or impossible to make use of the net via a QL. I've been surfing, emailing and more since 1994, all from the comfort of home, on my QL, all via a modem and local telephone calls to the ISP (Internet Service Provider), all put together without much effort.

The speed of pages' loading text-only, plus the familiar old green lettering, has been refreshing. As I write there is occasional thunder in the vicinity, but the deadline for "QL Today" is about 36 hours away. A simple QL setup is much less risked than even the moderately priced PC system on my other desk, and it is more than up to the task at hand.

To some this article will contain the obvious, while others will find it surprising.

* * *

This article is being written and researched on the QL, currently in Quill/Xchange (3.90l). QTPI is running in the background, occasionally connected to the 'Net. Hardware & system setup will be given later.

I have this week acquired important support group information for a non-Net-literate friend concerning a son with a chromosome defect. It was all via a 'Net search (Yahoo) on the QL. I have the last few days been out to the official World Cup 1998 site, and to the "Bridge of Wings" (BOW) site. BOW traces, nearly real-time, two American and two Russian pilots' air journey to commemorate a pioneering 1938 flight from Moscow across Siberia.

The final article will be delivered, via QL and email, to editor Jochen Merz in Duisburg, Germany, from East Tennessee in the US. Delivery will span six time zones, but likely will take minutes or less, far faster than in the past when I would've mailed a 3.5" floppy.

"All the World's a [QL] Stage" is a word play on a famous phrase from the writings of Shakespeare.

* * *

I have, recently or since 1994:

- 'Net-searched for, then visited the World Cup 1998 "Official Site" to retrieve articles on the Brazil-France Final. In my earlier 'Net article, I had reported on the America's Cup 1995, one of international sports' most patrician events.
- Also 'Net-searched for or 'Net-surfed to health organizations' web pages and to international aviation and friendship sites, as mentioned above;
- Both "lurked" and "posted" on various Internet discussion groups, including the world-spanning ql-users group, headquartered (physically, not just virtually) in Norway;
- Loaned my friend Sergey of Saratov (CAPATOB), Russia, my only computer at the time, the QL, so he could email to colleagues around the US and in Europe and Russia while he was visiting on USIA's "Business for Russia" program;
- Chatted real-time, simultaneously, with people from Asia, Europe, Canada and the US about commercial software packages;
- Sampled various "online community", "Internet portal" and search sites, such as Geocities, Yahoo, Metafind, and the WELL.
- Used or attempted use of the famous Internet tools Archie, and Gopher;
- Explored software libraries in Berlin and elsewhere in Europe;
- And, at the moment, I can still dial into the office over the phone lines and program or clean up accounts on various machines, again from my QL, with QTPI's VT100 emulation. That has, e.g., involved work on the International Energy Agency's (IEA's) Energy Technology Data Exchange (ETDE).

All the above has been from the comfort of home, all from my QL, all for the cost of local phone calls. It has been possible by obtaining an Internet "on-ramp" through a local Internet services provider (ISP), obtaining a comms program in the form of Jonathan Hudson's QTPI, and obtaining an appropriate cable for my US Robotics 14,400 modem plus an appropriately configured copy of QTPI from a dedicated QL'er (Don Walterman).

Some of the tales mentioned above are old, but I love citing them, since they may sound impossible on a basic QL setup.

It's been pretty much... well... Plug 'n' Play... modest apologies to M$, if they'll pardon the expression :-).

This is not a nuts and bolts or how-to article. Possibly for some readers I've already introduced a number of unfamiliar terms. Bill Cable, e.g., has written a good nuts and bolts and how-to series for NES-QLUG. One can cite various Internet articles in the QL literature.

Nor do I go into any of the current discussions on the ql-users mailing list (ql-users@nvg.ntnu.no), e.g., Aurora, Milan, and TCP/IP for the QL.

I cannot explain everything in a limited space. I'm simply relating adventures on a QL. Some how-to could follow in a later issue. If you are familiar with some of the net-surfing terminology related here, but in a different environment (say at work), realize it's possible on your QL. If you're not too familiar with browsers, the web, email, archie, and the like, realize there's a new world out there, and revel in the fact that it's accessible from your QL. What the flashy net browsers like Netscape and Explorer can do, you can do most of with a QL, QTPI, a local Internet services provider, and Lynx. (All the fancy graphics in Netscape just slow down getting the text from the Web).

* * *

Beginnings

My Internet adventures began in August of '94 when the real Internet discovered me. In one short time span I saw work colleagues work debut the US Department of Energy's Home Page on the World Wide Web and I stumbled across the first Internet services provider in my area. I decided I was behind the times and signed up for a dial-up account with US Internet, Inc. I started out with the famous US$8.95 blue-light special Everex-946 2400 baud modem and QTPI 1.35. About 1995 I graduated to a USR 14,400 fax/modem. The 14.4 is still hooked up to the QL.

Still in 1994, I was thrilled to send my first international email, from East Tennessee to a QLer in Sweden, and to get a reply. Now such actions occur without a thought: I communicate with Jochen Merz in Germany about the nature of this article without either of us even thinking about the fact that neither of us has picked up the phone, posted in "snail mail", or crossed the Atlantic.

When women write me via American Singles (www.as.org), or Match.com (www.match.com), and Single Booklovers (www.singlebooklovers.com), they usually include an EMail address. These URLs ('Net addresses) are all reachable with a QL, QTPI, etc. For now, when pictures get involved, I need the PC; however, I have heard of GIF viewers for the QL.

World Wide Web, Emailing & Downloading

A friend who works at a local restaurant asked for help: she and her husband have a son who has a condition she wrote down as "9P Trisomy." She wanted to get into the Internet and do a 'Net search via AOL (America OnLine) on the condition. She couldn't figure out how, and asked me to search for her.

In Yahoo, from the QL, moderate effort went from "9P Trisomy" to find pages on "Trisomy" and on parents' groups, research, and medical organizations. Real digging turned up pages on "Trisomy 9":

```
================================================================
Date: Wed, 12 Aug 1998 19:45:14 -0400 (EDT)
Subject: http:--www.geocities.com-Heartland-Acres-5287

Trisomy 9 International Parent Support
Providing Information To Families of Children with Trisomy 9.
Our group has grown from 13 in 1992 to almost 130 contacts
as of July, 1998!

This group is hosted by:     Bill & Alice Todd     Highland, CA
92346     Phone/Fax (xxx) xxx-xxxx  Email: atoddna@sprynet.com
Trisomy 9 Photo page     What's New     Medical Information
Helpful Links     [INLINE]
================================================================
```

Trisomy is a set of chromosome defects and resulting conditions.

A search for the world's largest, or one of the largest, sporting events turned up the Official World Cup web site and pages reporting the buildup to the Final between the expected winner Brazil and the underdog France:

```
================================================================
File that you are currently viewing
Linkname: FRANCE 98 - FINAL PREVIEW          URL:
http://www.france98.com/english/news/prev64.htm     Charset:
iso-8859-1 (assumed)      Server: Netscape-Enterprise/2.01
Date: Wed, 05 Aug 1998 00:24:58 GMT    Last Mod: Sat, 11 Jul
1998 20:44:09 GMT    Owner(s): None         size: 226 lines
Saint Denis stadium dancing to samba beat or French rock

Saint Denis Stadium will be dancing wildly to the Brazilian
samba beat or French rock late on Sunday as the World Cup
extravaganza finally makes its bow with either Dunga or Didier
Deschamps brandishing the famous golden trophy.
The 64-match, 33-day football ["soccer" for some Americans-
Ed.] jamboree with an estimated total television audience of 38
billion will come to a close as the celebrations of the winners
hold the attention of the sporting world. <...snip...>
================================================================
Sergey of Saratov kept in touch with his business colleagues
```

```
while at my home through my QL. After he was back home, we
exchanged email acknowledging the date June 22, the date in
1941 when Russia and the Soviet Union came under invasion. He
said, in part, "I am glad this dark period in both our nations'
history is past. Now necessary is contact between ordinary
people of both countries."

Two pilots, on July 4, took off in a single-engine taildragger
from Lebanon, TN, USA, three hours by car west of my home.
They would pick up two Russian pilots in Moscow and from there
commemorate--retrace--a pioneering flight by three Russian
pilots across Siberia in 1938.

The two American pilots had a laptop with them.  In an email
exchange before they got out of North America, they told me
international relations and contact was a major purpose of the
trip.

============================================================
Bridge of Wings
"The Commemorative Flight of the Rodina" 1938 - 1998
Keep up with the progress of the flight.
Click to view the Bridge of Wings Journal

On July 4, 1998, [the two pilots] will climb aboard their
Maule M-5, a single engine tailwheel aircraft, and fly it from
Nashville [Lebanon] to Moscow. From there they will retrace the
1938 flight of the Russian plane "Rodina" (which means
"Motherland").
Their mission is one with purpose. Sixty years ago three
Russian women set a world record when they flew non-stop from
Moscow to the southeastern tip of Siberia. [They] opened up the
route through the region and became a celebrated part of
aviation history. <...snip...>
============================================================
```

Discovering an interesting attribute common to all seven pilots is left as "an exercise for the reader."

Another important resource for the QL international community is the famous Thierry Godefroy web site, www.imaginet.fr/~godefroy. Perhaps no web site for a given field can be truly comprehensive. However, this one appears to [excerpts follow]:

```
============================================================
QL Web sites |QL FTP sites | QL newsgroups | Searches | Wired
QLers | QL BBSs | QLCF BBS | Available files on QLCF BBS |
Downloading | QL & compatible computers | QDOS & compatible OS |
Address book | Sign guest book | View guest book | Latest news |
Forum | ql-users mailing list | Chat room | General index
-----------
The Sinclair QL and QDOS compatible systems site

This site is dedicated to Sinclair QL and compatible computers
(Thor 8/20/21/XVI, QXL, emulators) and to QDOS, ARCOS, Minerva,
SMS2, SMSQ and SMSQ/E operating systems.

[LINK]   Sinclair QL forum   Updated !
[LINK]   ql-users mailing list archives   Updated !
[LINK]   The QLers' chat room
[LINK]   Latest news in the QL world.
[LINK]   Searches for QL sites on Internet.
[LINK]   List of wired QLers   Updated !
[LINK]   List of QL dedicated BBSs
File downloading   Updated ! <...etc...>

[INLINE] Other Web pages dealing with Sinclair QL and ....
* Generalist sites:    + Giorgio Garabello's page (in italian)
http://www.fortunecity.com/skyscraper/perl/357     <...etc...>
============================================================
```

The graphics encourage having a PC; nonetheless the site is quite viewable with even a close-to-original QL.

It is of course still possible to download web page finds through a browser [m]ail, [p]rint, or [s]ave command, usually to the ISP's machine, then transfer from there to the QL via the comms program's Zmodem or other ability.

The World Wide Web is hard to define, but easy to recognize when you see it. The World Wide Web Unleashed takes pages to try to define it. Find it and you'll recognize it.

All this while pounding the keys of my PC-keyboard attached to a QL with a Falkenberg interface. Really.

Coming up in the next part: Usenet and Discussion groups. ■

Using DBEasy's EASY_OUT_CUSTOM

Al Feng

A few years ago, I felt that I was just getting comfortable with the previous version of DBEasy [Wood and Wind Computing, RR3 Box 92, Cornish, NH 03745]. However, at that time I had modified both the "easy_out_cus" and "easy_out_line" procedures for printing simple disk labels which provided BOLD or italics output. In retrospect, I clearly did not have a full appreciation of the power of generating a custom output using the relational capabilities of the ARCHIVE database program.

A few years ago, someone asked me if there was an easy to use program which could generate a simple, monthly invoice. Because most accounting software is not simple to use (at least, initially), I thought DBEasy would be a good alternative if I could generate a monthly output from the individual "account" data. As the Fates would have it, in the few days that transpired between the original query and developing the prototype, I learned the need no longer existed.

The experience was not a lost effort -- at least, that's what I tell

TF Services

superHermes

A major hardware upgrade for the QL

- All Hermes features (see below for list) PLUS full 19200 throughput on ser1/ser2 not affected by sound
- IBM AT keyboard interface (plus foreign drivers)
- HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps
- THREE low speed RS232 inputs (1200 to 30bps) Driver for SERIAL MOUSE supplied. Other uses include RTTY/ graphics tablet etc
- THREE spare I/O lines (logic) with GND/+5V
- Capslock/scrollock LED connector
- Turbo/keylock connectors
- 1.5k user data permanently storeable in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software) **£90** (£92/£87/£90)
IBM AT UK layout Keyboard ..... **£22** (£24/£23/£27)
Serial mouse ..... **£11** (£13/£12/£14)
Capslock/scrollock LED ..... **£1** (£1.50/£1/£1.50)
Keyboard or mouse lead..... **£3** (£3.50/£3/£3.50)
High speed serial (ser3) lead..... **£4** (£4.50/£4/£4.50)

Hermes available for £25 (£26/£24/£27) **Working ser1/2 and independent input, debounced keyboard & keyclick.**

superHermes LITE

All Hermes features (see above) + an IBM AT keyboard interface only. Entry level superHermes.
Cost (incl keyboard lead)...**£53** (£55.50/£51/£53.50)

Minerva

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if requd). Send disk plus SAE or two IRCs

MKI..£40 (£41/£40/£43) **MKII..£65** (£66/£63/£67)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 including 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the Sinclair QL

NOW BEING SHIPPED

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent FLASH memory (ie there when the QL is switched off) which can be written to by the QL.

The software to access it is loaded automatically at power up/reset. It uses a directory driver written by Tony Tebby, and logic code from Stuart Honeyball. You can even load ROM images.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed (reading at over 1mbyte per second).

It is an extremely small and compact circuit board, and has hard gold edge connectors, eliminating contact problems and corrosion.

2 mbytes RomDisq.....**£39** (£41£37/£40)
4mbytes RomDisq.....**£65**(£66/£63/£67)
8 mbytes RomDisq.....**£98** (£100/£95/£99)
Aurora adaptor.....**£3** (£3.50/£3/£4)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface **16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)**
2 amp (for 8 relays, small motors).....**£40** (£43/£38/£44)
4 amp total (for motors etc).....**£45** (£48/£43/£50)
Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£23/£27)
Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required.. **£25** (£28/£23/£27)
Analogue Interface **Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temperature measurements, sound sampling (to 5 KHz), x/y plotting**..... **£30** (£31.50/£29/£30)
Temp probe (-40°C to +125°C)..... **£10** (£10.50/£10/£11)
Connector for four temp probes..... **£10** (£10.50/£10/£11)
Data sheets..... **£2** (£2.50/£2/£3)
Control software & manual (for all I/F).. **£2** (£2.50/£2/£3)

QL SPARES

Keyboard membrane..... **£12** (£12.50/£12/£13.50)
1377 PAL.....**£3** (£3.50/£3/£4)
Circuit diagrams.....**£3** (£3.50/£3/£4)
68008 cpu or 8049 IPC .....**£8** (£8.50/£7.50/£9)
8301/8302 or JM ROM or serial lead.. **£10** (£11.50/£10/£11)
Power supply (sea mail overseas).....**£12** (£17/£16/£21)
Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC,/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH! (No Eurocheques). SAE or IRC for full list and details

8 MAR 98

VISA

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 **Fax/BBS: 01442-828255**
tony@firshman.demon.co.uk **http://www.firshman.demon.co.uk**

MasterCard

myself! After all, there were some programming routines which I "worked out" (surely, others have done the same elsewhere) that I would not have bothered to attempt otherwise. The "refined" code is included in the following LISTing.
The standard DBEasy screen layout was used with the following sample labels and user input. Compare the generated output with the following record's data.

```
Using : INVOICE..DBF        Order : Natural
Date  : 94/01/05            Keys  : TxWA1001-5

ACCT. NAME  Wayne's Auto
ADDRESS  84 Granny Smith Road
more address
city  Pie Town
state  NM
zip  87327
description1  6 spark plugs
description2  5 qts oil 30wt
dateRECEIVED

Monthly FEE  5                itemized 1    5.88
PreviousDUE  17.33            itemized 2    6.45
0
```

In the following procedure, the "payment due" date is automatically calculated to be the last day of the subsequent month that the invoice is generated, with February's date adjusted to the "28th" of the month.

```
proc easy_out_cus
rem /*  use for invoicing  */
let day_due=30
let month_due=val(today$(1 to 2))+1
let year_due=val(today$(9 to 10))
if month_due=13: let month_due=1: endif
if month_due=1: let year_due=year_due+1: endif
if month_due=2: let day_due=28: endif
lprint
```

The fonts defined are for a 9-pin, EPSON compatible printer. Of course, you will want to substitute the appropriate codes for your printer. If you have a color printer, you can define the various colors as similar $trings. And similarly, you can define actual typefaces and scales, too.

```
rem /*  FX-80 printer control codes  */
let bold$=chr(0)+chr(27)+chr(69)
let bold_off$=chr(0)+chr(27)+chr(70)
let ital$=chr(0)+chr(27)+chr(52)
let ital_off$=chr(0)+chr(27)+chr(53)
let ff$=chr(0)+chr(12)
```

If you are a vendor, you will need to calculate the tax. The "rate" (0.675) is New Mexico's previous rate in effect when this procedure was written.

```
rem /*  tax rate */
let rate=0.0675
let Inpt3$=str(n1_+n2_+n4_+n5_,0,2)
let tax$=str((n1_+n4_+n5_)*rate,0,2)
let total=val(Inpt3$)+val(tax$)
```

Of course, how your actual output looks depends on the information you want to include. Regardless, of interest is how the printer's fonts are turned on and off.

```
rem /*  this is your header  */
lprint
lprint tab 32;bold$;"PLATYPUS Software"
lprint tab 29;bold_off$;"914 Rio Vista Circle  SW"
lprint tab 30;"Albuquerque, NM  87105"
lprint tab 35;ital$;"505 843-8414"
lprint ital_off$
rem /*  line spaces vary with your header       */
rem /*  adjust accordingly if using letterhead  */
rem /*  the following is account information     */
lprint
lprint tab 10;"Account Number:  ";key_$; tab 48;
"Current Balance:  $ ";total
lprint
rem /*  parameters for this are modified above  */
lprint tab 10;"Payment due by:
";month_due;"/";day_due;"/";year_due; tab 48;
"Amount Enclosed:  $ _____"
lprint : lprint
rem /*  account address set for window envelope  */
lprint
let i$=s1_$: if s2_$<>"": let i$=s2_$+" "+s1_$:
endif
lprint tab 10;i$
lprint tab 10;s3_$
if s4_$<>"": lprint tab 10;s4_$: endif
lprint tab 10;s5_$;", ";s6_$;"  ";s7_$
lprint : lprint : lprint
lprint "================[ Please return above
portion with your payment ]==============="
lprint
lprint
rem /*  customer retains this part */
lprint
lprint tab 10;"   Account Number:  ";key_$;
tab 50;"Billing date:  ";today$(1 to 6);
today$(9 to 10)
lprint
lprint tab 10;"Your check number:  _________";
tab 50;" Payment due:
";month_due;"/";day_due;"/";year_due
lprint : lprint : lprint
lprint tab 10;i$
lprint tab 10;s3_$
if s4_$<>"": lprint tab 10;s4_$: endif
lprint tab 10;s5_$;", ";s6_$;"  ";s7_$
lprint : lprint
```

The following method for fixing the two-place decimal output in self-adjusting column may not be the most compact or elegant sequence; but, it works. Since the working example does not utilize numerical fields "n3_" and "n6_" you should

note that there is no corresponding "AddOn3" or "AddOn6" value.

```
rem /*  to ensure two-place decimal output ...  */
let Inpt1$=str(n1_,0,2)
let Inpt2$=str(n2_,0,2)
let Inpt4$=str(n4_,0,2)
let Inpt5$=str(n5_,0,2)
let Ln1=len(Inpt1$): let Ln2=len(Inpt2$):
let Ln3=len(Inpt3$)
let Ln4=len(Inpt4$): let Ln5=len(Inpt5$):
let LnT=len(tax$)
let AddOn1=Ln3-Ln1
let AddOn2=Ln3-Ln2
let AddOn4=Ln3-Ln4
let AddOn5=Ln3-Ln5
let AddOnT=Ln3-LnT
let Blank$="        "
if AddOn1<1: let AddOn1$="": endif
if AddOn2<1: let AddOn2$="": endif
if AddOn4<1: let Addon4$="": endif
if AddOn5<1: let AddOn5$="": endif
if AddOnT<1: let AddOnT$="": endif
if AddOn1>=1: let AddOn1$=Blank$(1 to AddOn1): endif
if AddOn2>=1: let AddOn2$=Blank$(1 to AddOn2): endif
if AddOn4>=1: let AddOn4$=Blank$(1 to AddOn4): endif
if AddOn5>=1: let AddOn5$=Blank$(1 to AddOn5): endif
if AddOnT>=1: let AddonT$=Blank$(1 to AddOnT): endif
lprint tab 44;"   Previous balance:  $ ";
AddOn2$;Inpt2$
lprint
lprint tab 10;"> ";s8_$; tab 65;" $ "; AddOn4$;Inpt4$
lprint tab 10;"> ";s9_$; tab 65;" $ "; AddOn5$;Inpt5$
lprint
lprint tab 44;"Monthly service fee:  $ ";
AddOn1$;Inpt1$
lprint
lprint tab 54;"sales tax:  $ ";AddOnT$;Tax$
lprint tab 65;" --------"
lprint tab 44;"Current balance DUE:  $ ";total
lprint : lprint : lprint : lprint
```

If you are re-creating an invoice similar to this one, you will want to "end" the invoice by repeating the information contained in your header. In this example, the information is put in a single line.

```
lprint tab 27;ital$;"Thank you for your patronage";
ital_off$
lprint : lprint : lprint : lprint : lprint :
lprint : lprint
lprint bold$;" PLATYPUS Software";bold_off$;
"  914 Rio Vista Cir.  SW, Albuquerque, NM
87105  ";ital$;"505 843-8414";ital_off$
lprint ff$
endproc
```

Although the example given is not a standalone procedure, it can edited exclusive of DBEasy, saved, and then merged with the main program. While you may not have the need to generate an "invoice" from within DBEasy, I hope the preceding

gives you a good idea about how flexible both DBEasy and the ARCHIVE programming language can be.
The details of the sample invoice have "PLATYPUS Software" sourcing spark plugs and oil to "Wayne's Auto." Obviously, the appropriate data will be generated by your real world situation.
HAPPY TRAILS, AND COMPUTING, TO YOU ...

```
PLATYPUS Software
914 Rio Vista Circle  SW
Albuquerque, NM  87105
505 843-8414


Account Number:  TxWA1001-5          Current Balance:  $ 35.83

Payment due by:  4/30/96             Amount Enclosed:  $
_____


Wayne's Auto
84 Granny Smith Road
Pie Town, NM  87327


===============[ Please return above portion with your payment ]==============


Account Number:  TxWA1001-5          Billing date:  03/19/96

Your check number:  _________           Payment due:  4/30/96


Wayne's Auto
84 Granny Smith Road
Pie Town, NM  87327


Previous balance:  $ 17.33

› 6 spark plugs                                        $  5.88
› 5 qts oil 30wt                                       $  6.45

Monthly service fee:  $  5.00

sales tax:  $  1.17

--------
Current balance DUE:  $ 35.83



Thank you for your patronage



PLATYPUS Software  914 Rio Vista Cir.  SW, Albuquerque, NM  87105  505 843-8414
```

Time to vote

Joachim Van der Auwera

I think we have been discussing the use of directory separators for quite a while now in the ql-users discussion forum, expressing many opinions both for and against using different separators.
So please let me know your opinion about this issue, and I will try to publish the results (and maybe also some intermediate ones).

Question 1: Do you want to replace the underscore '_' by a different character as directory separator. If you choose to change, than please also state your preferred separator.
Question 2: Do you want to replace the underscore '_' by a different symbol as extension sepator. If so, what is the preferred separator.
Question 3: Should there be reserved characters which cannot be used in a directory or filename. If so, which characters?
Question 4: Do you think a limit of 36 characters on a filename (including extension) is reasonable, and if not, what maximum length whould you suggest.
Question 5: Do you think there should be a maximum length on name of a file including directory. If so, what maximum would you suggest?

Please send your vote to **vote@triathlon98.com** or via "ordinary" mail to PROGS. The results will be published on the mailing list and QL Today. Hoping that we can get some realistic results from this for future filing systems.

QLTOOLS 2.7q

Al Feng

Jan Venema indicated (May) that one of the utilities (QLTOOLS) was updated. QLTOOLS 2.7q can be is used to expedite transferring files directly from a QL formatted disk to the user's PC hard disk for use by the QLAY emulator.

As is the norm for DOS utilities, attempting to use the program without indicating the proper parameters will give you a screen with a series of options. I did not try this program in its original iteration, nor have I tried most of its options.

The syntax is simple ([command] [drive] [option switch]):

```
qltools a: -q
```

Indicating an individual file is meaningless as all the files on the disk will be transferred. As the documentation states, "all files on a QL formatted disk can be retrieved" which should be taken to mean that ALL files "will" be retrieved.
At first, I inadvertantly transferred ALL the files from a disk; so, it was necessary to delete some unwanted transfers.
Once you realize that QLTOOLS is not selective, you can pre-select files to be added to your QLAY.DIR by having only the specific files which you want to include on the QL disk being accessed. The QLAY.DIR will be automatically updated.

Once I realized that the utility worked as intended, I found it very useful for transferring all the DBEasy files that I had by first putting them on a separate disk and then using QLTOOLS.

Of course, since QLAY is only working directly with WIN1_ as a hardware device, you will have to modify directory calls appropriately, or as suggested in one of the readme texts, use the WIN_USE FLP (I tried this, and it works, but I found that using the CST utility [filed] was a better, long term solution).
Jan Venema previously indicated to me that his objective is to make using QLAY as simple as using a standard QL. This suggests to me that using the QLTOOLS is an interim step toward direct floppy drive access from within the emulation. He has done a lot of work in the program this Spring; and, the next version of QLAY may already be available by the time you read this.

qltools 27q is qltools 2.7 modified for use with QLAY/QLAYW. Copyright notice and qltools 2.7 manual are in the source directory. The source code for qltools is in there as well.
All functions of qltools 2.7 are preserved. One new optoin is added: -q. All files on a QL floppy can be retrieved and stored on a PC directory. A 'qlay.dir' file is created.
■

Internet Access from QDOS

The latest version of the uqlx emulator includes the IPDEV option giving full Internet access from QDOS.
uqlx/IPDEV provides Internet device drivers (tcp_, udp_ and sck_) written by Jonathan Hudson with assistance (particularly in the integration and testing phases) from the uqlx maintainer, Richard Zidlicky.
The drivers provide limited Internet access from SuperBASIC (one example is an Internet news reader in SuperBASIC); with complete access (via a c68 'socket' library) available from programs in 'C' (or assembler). The socket library provides a complete BSD socket compatibility and makes porting Unix internet/TCPIP programs 'trivial', at least in the networking area.
A number of support and application packages are available, the socket library, an ftp client and the lynx Web browser. lynx 2.7.1d provides many services including full internet ftp, news, and WWW access.
The source for the uqlx drivers (QLip.c, QLip.h) are part of the uqlx package; however these particular files are freely distributable for non-commercial use, with the hope that they might be useful to the authors of the other 'host-multitasking' emulators (QLAY and Qemulator).

The following resources may be useful in obtaining internet access from QDOS.

The UQLX home page (Richard Zidlicky) supplies the uqlx source:
http://www.geocities.com/SiliconValley/Bay/2602/uqlx_main.html

The Dead Letter Drop (Jonathan Hudson):
http://www.jrhudson.demon.co.uk/index.html
supplies the application files

`qlsocket.zip`	the 'socket' library and examples (C and SuperBASIC).
`qlftp.zip`	ftp client
`qlynx271d.zip`	lynx web browser

More information from Jonathan Hudson or Richard Zidlicky.
■

Operating Systems, forgotten aspects

H.P. Huyg

1. Introduction

A computer system consists of a number of components. There is at least one 'engine', a certain amount of directly accessible memory, external memory may be permanently attached or not, things to communicate with peers (network), or the outside world, like printers, keyboards, screens and the like. On the other hand there are the users of the system. An Operating System (OS) could be renamed as a resources management system. The early computers had no OS, just a collection of 'drivers', bits of software to handle the tedious logic to read punched cards or papertape and to output information to a printer, punched cards or papertape. Drivers still exist, they are the interface between the engine and the set of peripherals, most unlikely to be punched cards or papertape though! So this is one function for the OS. Life has become much more complicated, mainly for three reasons: the ability of modern equipment to serve more than one user or application simultaneously, the introduction of common resources and the usage of certain programming techniques. In fact, one could say that it is especially the usage of common resources making an OS complex and not seldomly unstable. The screen, keyboard, internal memory, disk memory, printers and processor power are all resources to be shared in one way or another. This article will investigate one such resource, namely the hard disk, and show that the expression: Resource Management is a misnomer. Another article will address the issue of internal memory assignment in an operating system.

2. File Attributes

Any hard disk has a finite space capacity. If the unit of information is a character (often 8-bits, normally called byte, or sometimes octet), then each character is either available or it is reserved, the latter meaning that its usage is restricted to a specific (set of) application(s), or indeed to the Operating System. Very often there is a 'pool' of available characters. The restricted characters are addressable by a name plus a ranking sequence number. A name is a hierarchical thing, often consisting of:

- a unit identification,
- likely one or more directories,
- a filename,

the collation of it being unique.

Most often the directory structure is part of the OS. For completeness' sake, it should be noted that an application can read or write a series of characters starting from any rank within a file (usually with a maximum). Most environments don't like characters to be read if they have not been written before.

Files have attributes, depending on the OS they can be:

- the starting position on the disk,
- the number of characters it effectively occupies,
- the date & time it has been created, modified, accessed,
- the projected lifetime,
- the nature of the file (is it text, a program, control information, a database, ...),
- access rights,
- maximum possible space, ...

A number of these attributes can exist as well at the directory level. It is unfortunate that the set of attributes of files/directories is decided by the makers of an OS and frozen, until such moment that, for whatever reasons, a new fixed set appears. Wouldn't it be nice, if, when installing an OS, one could decide which of the attributes should be provided, with the necessary 'hooks' (some mandatory, some optional)?

3. Filesize

Well before the advent of microprocessors, the OS used to be written by the hardware supplier (notable exception: UNIX). Possibly because of this, there has been a subtle change at one stage about the allocation of space on hard disks. In the good old days the application asked for a certain amount of space and that would be the maximum filesize. If it was available, fine, if not, tough luck, the application then refused to continue. In a way this technique created quite a few problems, especially with year-end work, where, often, volumes of data were a multiple of the 'normal' workload. One had to plan ahead.

This has been changed. Most Operating Systems give 'slices' of space on demand, the limit being a full disk. This led to:

- the abolishment of the data space planning operation,
- huge increase in bigger disks sales/rentals,
- the establishment of data space rescue operations, usually as a post-mortem action.

Everybody was happy, except the Financial Comptroller of the mainframe users. Please note that it is not so much the change in itself that there is an objection to, (in a development environment this is one bother less) but the fact that there is no way to have fixed data assignment to coexist with the current techniques, they are not mutually exclusive!

So, the OS does not manage

the disk space, it is giving away as long as it can, that is playing Santa Claus, and, what is worse, it makes the planners work very difficult, if not impossible. By the way, the author seems to remember that in UNIX based systems a user can be assigned a maximum amount of space, that is something.

4. File Usage

File and/or Directories should have an owner, a body deciding what can be done to them and by whom. Being a bit more specific: any (permanent?) File should carry with it the information telling which application(s)/program(s)/user(s) may:

- consult it, and/or
- create it, and/or
- update it, and/or
- transfer it, and/or
- execute it.

This looks heavy, but as it can be imposed at any level (one of the advantages of a hierarchical directory structure), it is up to the user to decide to which level of detail the working environment has to be controlled.

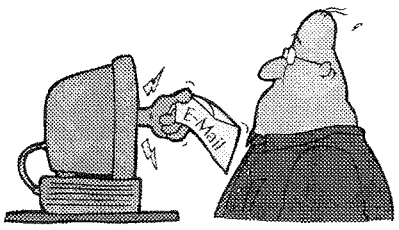
It will allow the user for example:

- to declare whole disk units or directories as 'read only',
- to confine new (and outside) applications to their own 'working grounds' with very specific 'gateways' to their outside world.

Of course the situation in some current, dominant, Operating Systems that everything is accessible, etc, by 'any' program, even from outside the computer system itself, is perfectly implementable, it is just one of the many possibilities.

It is the author's contention that Operating Systems not permitting an adequate control of file-usage should not be used in a business environment.

■



Letter-Box

Don Walterman writes:

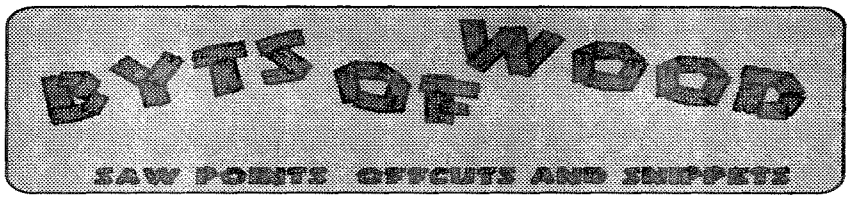
Just an observation on the Pandora case article in the last QL Today....

I built my Aurora system into a Pandora case and haul it all over the place (including the 1300 mile round trip to the Bedford QL show)....

When I assembled my Pandora, I didn't like the lack of support for the QPlane. I took the "L" shaped part of Roy Wood's bracket and mounted the QPlane to it. That required 2 holes in the QPlane and 2 holes in the bottom of the Pandora. The bracket needed 3 washers to space it up high enough. That keeps the QPlane from bouncing around and wearing out the lower connector. I would suggest The Pandora implement that or people just buy a Bracket from Roy Wood....

George Gwilt writes:

I agree with Mr Tanner (Letter-box May/June) that the options of the user of a program should not be restricted by the programmer. I was surprised therefore to read that GWASS was guilty of this. The fact is that current versions of GWASS examine the filename presented and use it if possible, only adding DATAD$ or PROGD$ if this fails (an extension I owe to Dave Walker). This means that the options in GWASS are extended, not restricted. I myself, until fairly recently, always used the full name for any file to be assembled, so the values of DATAD$ and PROGD$ were to me totally academic. However lately I have been using GWASS, in batch mode, (an addition suggested by Simon Goodwin) to assemble a set of programs which I do by the use of a SuperBASIC procedure. This procedure puts together the required filenames but without the device or directory so that I can use the same call to assemble two different sets, one in RAM2_ and the other in RAM3_ merely by altering the value of DATAD$.



Its amazing what you get used to isn't it? I have used QDOS for over 14 years now and I only changed over to SMSQ/E comparatively recently but I have already forgotten which bits are peculiar to SMSQ/E. I apologise to some of our readers who tried my little Basic procedure in the last issue and who only have QDOS. I neglected to mention that you should have opened a window and then used the OUTLN command to set the Outline of that window in order to get SuperBasic to correctly display the menu and give points of reference for the Pointer Environment. Also Line 170 mysteriously found itself on the same line as 160 and this caused some problems to readers too.

How Long?

There has been a lot of comment on the ql-user list this month about the filing system and how people would like to see a change in it. Many of the people seem to miss the point that the contributors to this list are not typical QL Users and so any concensus reached by these subscribers is a very one-sided view. I would like to throw this discussion open to others and so I am including some of their suggestions here.

The crux of their arguments seems to be that they want to have longer file names than the current 36 character limit and they would prefer to have the file path removed from the filename. Since I am not a programmer myself (at least not any more than just tinkering around) I cannot comment on the validity of many of the arguments used except to say that as a user I can see no reason to change at all.

I can understand that people want to be able to give more meaningful names to some of their files so the only point at which I would go along with some of this is to increase the number of characters available for this function but almost anything else would mean that all of the current programs would have to be re-written and, as we all well know, that is very unlikely.

There are two distinct reasons the changes are being called for. One is that many of the 'C' programs that are ported from other systems do not work without a longer filename system. This is a problem inherent in the way in which they are written but it does lead us into a situation in which each of these programs call on many other files and 'addons' in a similar way to which the Windoze style programs do. Personally speaking I am not sure that this is a path that I wish the QL to travel down. I enjoy its simplicity compared with other systems and part of the joy of the thing is to be able to write short procedures to do little tasks that are not otherwise provided.

The other reason people want a change in the filing system is to allow the longer Internet names to be available when/if a TCP/IP stack becomes a reality on QDOS/SMSQ machines. This is a bit a of a problem because the whole internet is based on a system which has no hard paths. If you call a file all you see at the program end is the filename itself and the path is handled separately. That way, if the program has a call to another filename in it, it is assumed (unless otherwise stated) that that file is in the same path.

Jochen and I have discussed this problem and he says that there is no reason why the program which is doing the browsing cannot handle these problems directly thus leaving the original filing system alone to perform as it always has.

This said, many of the people on the ql-users list who were the prime movers in these debates will not be reading this because, as far as Jochen and I can tell, they do not subscribe to the magazine. It seems that some of the people shouting the loudest about this and demanding changes to the system are not those actively supporting the system and the magazine. I suppose they would not be prepared to pay for the changes they want either.

The Old 'Why re-invent the wheel?' Chestnut

Of course this does mean that we have to write a browser and a lot of people come up with the cliche above. As usual the use of cliches like this masks the real core of the argument. The very use of cliches in discussions is usually the same as saying 'I can't be bother to put any real thought into this so I will just trot out a few phrases that someone else said'.

There is a very good reason to re-invent the wheel. You might just make a better one. If you consider that the original wheel was probably a round stone or circular piece of wood the

people who re-invented it with a pneumatic tyre on it saved us all from a very bumpy ride. There is no reason to abandon the wheel concept but equally no reason why we should all look at it from the same angle - especially when, in this case the angle is from directly beneath the wheel of a UNIX juggernaut which is threatening to run us all down.

If you have any thoughts on this please let us know. Maybe I am completely left field here you are all thirsting to write files called

win1/data/documents/blathering/file/I/wrote/last/week_txt

(Oh, I forgot to say they want to change the separator as well)

The Code is out there...

There do seem to be a lot of conspiracy theories around and among these is the one that Jochen and Tony Tebby deliberately hide away all the code for their products so that no-one else can find out about it. This kind of concept is plainly nonsense but I somehow seem to have to keep repeating it to people. The problem springs from two different camps.

One of these is a small number of people who object to contacting Jochen for a code to use in the level 2 config blocks. The idea behind this is that each program that uses these blocks has a definitive id so it can instantly recognise which set of config data belongs to which program when using the updating facility. This definitive id is kept by Jochen in a master list and anyone who wants to use a level two config block registers this with him to prevent conflict.

You could, of course, not bother to keep this list and use some other criteria such as the name of the file or something similar. Of course we cannot be sure that every programmer knows which programs are being released by whom, especially in the Public Domain, and it is entirely possible that someone else has used the same name. Then you would get a conflict in the menuconfig_inf file and maybe even the wrong data being stuffed into the program causing at the least wrong operation and at worst a complete corruption of the program. So, do you still think it is not a good idea to keep a central list??

The second school of thought is that the above mentioned people also hide away all of the code for their programs so we cannot get to it. Of course Microsoft give away all of the source code for their products don't they? LINUX and many other free systems do hand out the code with the program but they are essentially free and therefore it is irrelevant that the code is available. Believe it or not Jochen, Tony and others do have to earn some money to keep everything going and, if they were not paid for the work that they do, they would have to do something else to earn money and there would be very little QL work done.

Some of these complaints refer to the non-release of the codes to access the 'Thing' system and other parts of the programs that could be used within the system. This information is available in many ways. Some of it is in Jochen's excellent 'QDOS Reference manual' - yes you do have to pay for it but then it takes him a lot of work to produce it and print it so that is justified. If you want specific information you can fax or call Jochen for it and he will give it to you. 'If you don't ask', as my old mum says,'you don't get'. Is there a problem with this?

Flying The Standard

Another subject that came up during recent user group exchanges is that of standardisation. I can remember my first encounter with Tony Tebby at one of the Eindhoven meetings. I had reported a bug in a piece of his code and he came over to my machine to try and find it. I had forgotten to bring my mouse mat with me and the mouse was not behaving very well on the hard table surface so my first impressions of the man were similar a small boy with a toy car. Running it backwards and forwards on the table but instead of going 'Vroom Vroom' he was saying 'What's wrong with this?'. He then pressed an altkey combination and was shocked that QMON did not pop up.

This leads me into the thorny question of standardisation. Many QDOS/SMSQ 'Power Users' relish the fact that their systems are set up in an idiosyncratic way and, when quizzed about the key combinations and subdirectories that they use, have very good reasons for them. In Gatesville the user is given a choice of where to put the files that relate to each program but most just click on the 'Yes' button and take what is thrown at them. Of course this is the simple solution and the program goes ahead and installs itself into a standard configuration which almost anyone who knows anything about PCs can get into.

In QDOSland however we all copy our programs into subdirectories of our own choosing (and I have met some people with no subdirectories at all), we create our own ALT keys and, when someone else comes to our system they find themselves lost. The question is what happens when a user cannot work out for himself what to do to integrate a program into his system. The old days of 'pop the disk into the flp1_ and reset' are not the 'old days' for a lot of people out there and those of us who think we know it all should stop for a while and think about why we have lost some of our user base. It may not be because their are not 256 colours but because the PC makes it easy for them to get going.

Where these people struggled to configure Text 87 or get a QPAC II boot file up and running so they could write a letter or copy a file on their QL the PC gives it to them on a plate. Is this not, therefore, time to consider these people and think about a standard setup facility for our programs? When PROGS produced this for ProWesS it was a step in the right direction even if some of the choices were a bit ambiguous and it did not really integrate itself into the system in a seamless manner.

Steve Hall has been looking into writing a QPAC II installer but we really need some concept of what a standard system comprises of and which files need to be loaded for it. The resulting installation program would use QPAC II as a starting point but would also be able to store and install executables and resident procedures on a hard disk or RomDisq as well as writing the boot file that calls them. We would welcome any feedback on this subject so write to Steve Hall at Qbranch.

Now C here

We increasingly divide into two camps. The people who stick with the QL because they are not willing or don't feel they are able to learn another system and those who are here because they find the system liberating and interesting to work with and program. In the second camp there are a lot of people squabbling about whether 'C' is a better language than assembler or the arguments described above. Most of this is very unproductive and the existence of these high profile disputes in Quanta and the other public forums leads some users to abandon the publications and newsgroups. Once this has happened these people lose touch with what is happening and the news of new programs and hardware and we lose them.

'C', of course, is a highly portable language and can be used on a number of different platforms by just recompiling the original source code with the relevant compiler and library. This comes at a price because the resulting code is less compact and runs slower than an equivalent piece of assembler. When you run this on a 266 MHz + processor the speed problems are not noticed but then we are operating on 25MHz QXLs at best at the moment so speed of processing is a definite criteria that should be taken into account.

What the tinkerers want is to play around with code and experiment with porting programs and that is by no means something we should discourage but what the users want is a system that runs well and 'does the job' and that is also part of the whole picture. The tinkerers, generally speaking, do not buy a lot of programs or hardware whereas the users do. If we lose this commercial aspect to the 'free publicly distributed program code' world we also lose the input of

the people who provide a lot of our system and the whole thing collapses. The tinkerers do not write for the magazine, or put up the money to publish it - this comes from the users and the cash input that they put in. Without this input we are all sunk. We need both worlds to keep QDOS/SMSQ alive and the last thing that I would like to see is a war between the two camps.

QD or not QD

No question really - at least not for me. There are a lot of text editors around from the free QED to The Editor, Master Spy and a whole host of others. I came across QD early in my conversion to the Pointer Environment and now I could not operate without it.

I am writing this article in a beta test version of the new incarnation of this program and this is a real leap in usability. Jochen has put a lot of work into rewriting sections of this program and has added a whole swathe of new features and ideas. My favourite is 'GOTO' blocks that appear in the toolbar. You can now move the pointer into the toolbar and click on 'LABEL' which offers you a menu of the first word in each line (or the line number if it is a BASIC program) and allows you jump straight there. Two other items in the toolbar allow you to jump to a Procedure or a function. Other nice touches include popup labels for the items on the toolbar, a new print menu that allows you to write a basic filter to print with and stepping arrows for most of the items in the menus that require numbers to be entered. Work has not finished on this new version of the program but it is already a vast improvement over a program that I was already very impressed with. The only other editor I ever use is Master Spy because that allows me to make changes to machine code files and other executables - maybe Jochen could add that too.....

Geoff puts a Spell on You

Another little program I have been beta testing recently is Geoff Wicks Spelling Crib. This is a useful little utility which I believe that Geoff may release soon. You can pop it up over any word processor and enter part of a word with a '/' indicating the part you are not sure about. It will then give you a list of words that fit that pattern. Very nice little program Geoff.

■

The QL Show Agenda

Sunday, 4th of October 1998: The BYFLEET WORKSHOP

The show will be on between 9:30 and 5, in BYFLEET VILLAGE HALL.
BYFLEET is just inside the M25, on A245. From M25 Jn 10 take A3 towards London then A245 towards Woking, and turn left (south) into Byfleet at the first or second roundabout past the A318 junction.
From M25 Jn 11 take A317 towards Weybridge then A318 to Brooklands/Byfleet. NB A318 now goes through Brooklands racetrack. Turn right (west) at A245.
Non-M25 is via A245 from west or east (or A3).
The hall is on the (old) High Road, western end, just south of A245. It is about 12 mins walk from Byfleet & New Haw station, on the main Waterloo - Woking line.
Queries? Ken Bain (SQSG Sec) 01932 347 432, pre midnight, email kenb@bcs.org.uk

Sat. & Sunday, 10th and 11th of October

International QL Show in Heidenreichstein, Austria (near the Czech border).
The venue is the Gasthof-Restaurant Nöbauer, A-3860 Heidenreichstein, Schremser Straße 28 (Tel. 0043(0) 2862/52237 or 52746). A room the size of the one in Salzburg (after moving) is available. The Gasthof provides 22 rooms with 22 beds. You can also find other accommodation in the same town.
As usual, there is also a "tourist" program for the visitors: a 1-2 hour ride on an old-fashion train (paid for by the Vienna QL user group), followed by a dinner with local specialities. On Sunday, it is possible to visit the moor and one of the best-kept water-castles. A visit to a local glass-manufacturer should be no problem.

Saturday, 31st of October 1998: Eindhoven - International meeting

Again, as usual, between 10 am and 4pm in the St. Joris College. We expect all dealers to come, and of course lots of international guests will be invited.
Probably THE date on the continent for Autumn.

Sunday, 8th of November 1998: Portishead (Bristol) Workshop.

Somerset Hall - Portishead. 10:00am to 5:00pm. Well visited, well organised workshop. Well worth attending!
Directions: M5 motorway, leave motoray at junction 19 and follow signs into Portishead. You will pick up a sign saying "Quanta" or something similar. Approaching the centre of Portishead you will see a Coop supermarket and a large car park. Turn left at the lights and pull into the car park on the right. Go through to the shopping precinct and to the Somerset Hall entrance and up the stairs. Look forward to seeing you there.

Saturday, 14th of November 1998: Bingley Hall